Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning for Internet of Things

Bimal Ghimire, *Graduate Student Member, IEEE*, and Danda B. Rawat, *Senior Member, IEEE*

***Abstract*—Decentralized paradigm in the field of cybersecurity and machine learning (ML) for the emerging Internet of Things (IoT) has gained a lot of attention from the government, academia, and industries in recent years. Federated cybersecurity (FC) is regarded as a revolutionary concept to make the IoT safer and more efficient in the future. This emerging concept has the potential of detecting security threats, taking countermeasures, and limiting the spreading of threats over the IoT network system efficiently. An objective of cybersecurity is achieved by forming the federation of the learned and shared model on top of various participants. Federated learning (FL), which is regarded as a privacy-aware ML model, is particularly useful to secure the vulnerable IoT environment. In this article, we start with background and comparison of centralized learning, distributed on-site learning, and FL, which is then followed by a survey of the application of FL to cybersecurity for IoT. This survey primarily focuses on the security aspect but it also discusses several approaches that address the performance issues (e.g., accuracy, latency, resource constraint, and others) associated with FL, which may impact the security and overall performance of the IoT. To anticipate the future evolution of this new paradigm, we discuss the main ongoing research efforts, challenges, and research trends in this area. With this article, readers can have a more thorough understanding of FL for cybersecurity as well as cybersecurity for FL, different security attacks, and countermeasures.**

***Index Terms*—Cybersecurity, data offloading, federated cybersecurity (FC), federated learning (FL), machine learning (ML).**

I. Introduction

With the explosive rise of connected devices, such as personal digital assistants (PDAs), Internet of Things (IoT), wearable medical devices, and others, an unprecedented amount of data is being generated every fraction of time. The immense volume of data has provided a better opportunity to utilize the machine learning (ML) model in general and deep learning (DL) in numerous domains [1]. Today, ML has made its way even to our everyday lives. From the small hand-held devices, IoT sensors, and cyber–physical systems (CPSs) to big companies, such as Facebook, Google, Amazon, Netfilx have been applying ML for their applications and services. Amazon Web Services, Google Cloud, and Microsoft Azure, just to name but a few, are some popular ML services [2], where models can be deployed and used at scale. ML has been inevitable not only to improve user experience and business modeling but also to detect cyber threats and cyber attacks and prevent them. Today's world heavily exists on data and maintaining its integrity and privacy is of utmost priority. Sensitive data related to individuals, organizations, and governments need to travel from one point to another through a communication link. Traditional methods of combating cybersecurity issues mostly protect devices only after the occurrence of specific types of attacks. However, the types and patterns of attacks in today's cyberspace have changed drastically. Attacks using polymorphic viruses keep on changing their signature and are difficult to detect and predict. So, the ML approach of detecting and predicting threats, anomalies, or any kind of security breach in cyberspace and taking corresponding countermeasures is gaining so much attention in recent years. Forming a centralized learning model by sharing local training data has already proven to improve the learning model's performance [3].

There are multiple models in practice for ML-based cybersecurity each with its advantages and disadvantages, namely, centralized, decentralized, and federated [1]. The federated learning (FL) model for cybersecurity is a recent addition among these models. We discuss all these models in the subsequent sections. Moreover, FL has been explored for its applicability in several areas, such as smart city [4], healthcare [5], recommender system [6], wireless communication [7], edge network [8], electric grid [9], vehicular ad hoc network [10], and many more. The FL framework inherently supports security and privacy (compared to the centralized learning framework) as data generated in an end device does not leave the device. The useful device data are used locally to train the learning model running on the device in a distributed manner. Only the updated parameters are exchanged between an end device and the cloud server. However, this approach still exposes several security threats. So, this survey primarily focuses on the security aspect of the application of FL. The FL framework offers promising potential to improve security and privacy, but for the success of it, the issues that hinder

Manuscript received October 25, 2021; revised January 11, 2022; accepted February 6, 2022. Date of publication February 10, 2022; date of current version May 23, 2022. This work was supported in part by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) at Howard University under Contract W911NF-20-2-0277 with the U.S. Army Research Laboratory; in part by the U.S. National Science Foundation (NSF) under Grant CNS/SaTC 2039583 and Grant 1828811; and in part by the National Nuclear Security Agency (NNSA) Grant. *(Corresponding author: Danda B. Rawat.)*

The authors are with the Department of Electrical and Computer Science, Howard University, Washington, DC 20059 USA (e-mail: bimal.ghimire@howard.edu; danda.rawat@howard.edu).

Digital Object Identifier 10.1109/JIOT.2022.3150363

the performance of FL must be addressed. In this regard, we also discuss existing works that address such issues such as the accuracy of FL model, latency of communication, data distribution, and resource constraint of distributed devices.

Due to the increasing complexity of software and communication interfaces, IoT and cyber–physical devices are more vulnerable to various kinds of attacks. Cybersecurity breaches in such systems are likely to incur several privacy and security issues. Appropriate safety measures and effective and robust cybersecurity solutions are mandatory to combat any threats or attacks. Below, we outline some common security risks associated with IoT and CPS where ML algorithms rely on data collected from such IoT/CPS systems.

Attacks on IoT/CPS Devices: Hackers can easily crack the passcode of devices with a brute force attack and manipulate bluetooth connectivity of such devices to leak private information, manipulate data, and/or gain control.

Attacks on Cloud-Based Networks: IoT and CPSs need to process a huge volume of data stored in the cloud frequently. These devices use different mediums of communication, such as Wi-Fi, cellular network, etc., to send and receive data to and from the cloud. These communication mediums are vulnerable to attackers and attackers in the middle might intercept and forge the data being exchanged.

Malware: Like any other connected device, IoT and cyber–physical devices are also susceptible to malware attacks.

Vulnerable Sensors: IoT and CPS devices are equipped with a wide range of sensors to monitor and support the systems. These sensors are vulnerable enough to be attacked by adversaries to cause security and safety threats. Even major sensors, such as global positioning system (GPS) signal, light detection and ranging (LiDAR) signal, inertial measurement unit (IMU) data, and so on can be compromised, which cause serious threats to the devices.

Network Attacks: Every device or endpoint in IoT and CPSs is a part of the network attack surface. Attackers can target the endpoints of the network and gain access to the network to control and compromise the whole system. Protocols, such as WiFi, Bluetooth, and GSM, allow external devices to connect and communicate with various sensors. These protocols contain bugs and are vulnerable to be exploited by attackers.

Firmware Attacks: In this form of attack, an attacker provides a malicious firmware update to a device by which he/she can get direct access to the whole system.

There are already several surveys (e.g., [1] and [11]–[15]), which reviewed FL and highlighted its taxonomies, methods, advances, applications, challenges, and more. However, our work is different from others since it presents the study about FL for cybersecurity and cybersecurity for FL in the CPS/IoT environment. The successful adoption of FL for the IoT environment hugely depends on several performance metrics, which are also reviewed and presented in this article. To combat various kinds of cyberthreats, an intrusion detection system (IDS) and intrusion prevention system (IPS) should be in place. Such systems must learn about the existing cyberthreats globally and even need to be proactive to detect and predict new and emerging threats. Collaborative learning framework of FL is suitable for such tasks. To evaluate security solutions properly, there have been significant efforts to create real data sets for more than two decades. This survey also highlights such works and discusses most of the data sets used by the research presented in this survey. We also discuss some popular data sets used in federated setting to evaluate federated model's performance. A shift in this new architecture of learning has introduced some novel attacks, such as poisoning and reverse engineering, and we also discuss research works that address these attacks. In this survey, in addition to discussing several recent research works in the field of FL, we also present ML algorithms and technologies applied by those works. The aim of this survey is to assist readers to choose a particular research direction with overall information. Specifically, the main contributions of this article include as follows.

1) We present a detailed study on federated models for ML and cybersecurity by categorizing them into two parts. The first part discusses the FL and its application in cybersecurity and the second part discusses cybersecurity for FL. Our study mainly focuses on the IoT/CPS environment.
2) As successful adoption of federated models for the IoT environment hugely depends on several performance metrics, we also present those metrics, challenges associated with them, and the potential solutions in this article.
3) We also present and discuss data sets used by the surveyed articles to evaluate their model's performance.
4) We have also presented cyberattacks, such as parameter poisoning and reverse engineering in FL.
5) We summarize security attacks and countermeasures and the addressed performance issues in federated models for IoT networks in a tabular form for a side-by-side comparison.
6) We present a discussion of research challenges, open problems, and recommendations for federated models that are needed to be addressed to realize their full potential.

The remainder of this article is organized as follows. In Section II, we discuss and compare different types of ML models. Existing recent works related to using FL as a tool to secure IoT environments and that related to making the FL framework secure are discussed in Section III. Some research efforts to address the issues that affects the performance of FL are presented in Section IV. In Section V, we highlight ML algorithms, technologies, and frameworks and in Section VI, we discuss data sets used by the surveyed research, respectively. Some open challenges and future research directions in FL for the IoT domain are presented in Section VII. Finally, we conclude our survey work in Section VIII. Full forms of various abbreviations are given in Table I.

II. Overview of Federated Learning and Federated Cybersecurity Model

In this section, we first present a brief overview of different types of learning models and then elaborate more on FL along with its challenges. Finally, we present a

TABLE I
ABBREVIATIONS AND FULL FORMS

Symbol	Full Form
CNN	Convolution Neural Network
GRU	Gated Recurrent Unit
SAE	Stacked Autoencoders
AWID	Aegean Wi-Fi Intrusion Dataset
MNIST	Modified National Institute of Standards and Tech.
Cifar10	Canadian Institute For Advanced Research dataset
LSTM	Long Short-Term Memory Networks
SVM	Support Vector Machine
VGG11	Visual Geometry Group
KWS	keyword spotting
NS3	Network simulator 3
DNN	Deep Neural Networks
DRL	Double Deep Q Learning
EV	Electric Vehicle
MLP	Multilayer Perceptron
KNN	K-Nearest Neighbor
SOHO	Small Office or Home Office
ADS BC	Anomaly Detection System Blockchain
RF	Random Forest
ECC	Elliptic Curve Cryptographic
IDS	Intrusion Detection System
SDN	Software Defined Network
NFV	Network Function Virtualization
WAN	Wide Area Network
DTN	Delay Tolerant Networking
IIoT	Industrial Internet of Things

federated cybersecurity (FC) model useful to protect the FL framework.

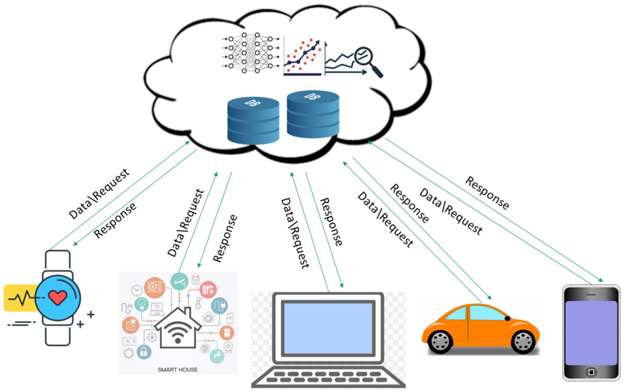

Fig. 1. Centralized learning model for IoT.

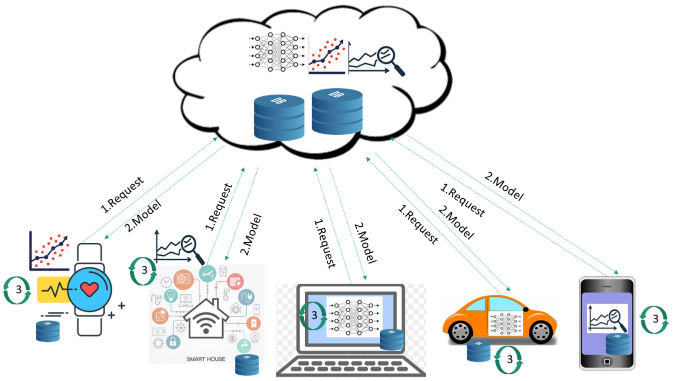

Fig. 2. Distributed on-site learning model for the IoT.

A. Typical Types of Learning Models

Approaches to combating cybersecurity issues have been changing continuously with the needs. To cope with the unprecedented growth of heterogeneous connected devices and a tremendous volume of data and traffic generated by them and the development of sophisticated tools to create polymorphic malware and other threats, ML has been an integral part of cyber defense mechanism in recent times. This section discusses three different ML-enabled models with their advantages and disadvantages.

1) Centralized Learning Model: This model uses the cloud-centric architecture (e.g., [16]–[19]) where data sent from end devices is centrally stored and processed in the cloud. In the cloud, data are analyzed, features are extracted, and then models are built on top of the stored data. Models are accessed by the end devices sending requests through an API. This approach offers significant advantages but carries some serious issues. One big advantage of this approach is that the cloud offers a huge repository so that storing huge volumes of data sent by all the clients will not be problematic. Another advantage is that the cloud is mostly equipped with high-performance servers. These benefits facilitate the building of better-trained models. Moreover, cloud services are best protected by service providers for any security breaches or attacks. Offering such great advantages, this approach has serious concerns over privacy, security, and latency. All the data needs to travel to the cloud through insecure communication links make the data vulnerable to being hacked by adversaries. All the private data generated by the devices are stored in the cloud, which raises big privacy concerns. Furthermore, the central authority or the cloud service provider has all the control over the model and data. Additionally, as data need to travel to and from the cloud, latency and bandwidth costs could be big issues if the communication distance between device and cloud is high. The working model of centralized learning is shown in Fig. 1.

2) Distributed On-Site Learning Model: In this approach of learning model, a generic or pretrained model is distributed by the server to all the devices or clients beforehand. After this, each device personalizes the model with training and testing with local data and learns the data generation process. Such a learned model enables predictions and inferences from live-streaming data generated by the device [1]. The big advantage here is data generated by the device stay locally, thus eliminating security, privacy, and latency concerns. The main downside of this approach is that IoT devices are relatively heterogeneous and weak in terms of memory, computation, and battery power. These devices are not suitable for the intensive computation required while using the model [20]. Furthermore, the locally running model lacks global updates or knowledge about new and emerging security threats. The working model of distributed on-site learning is shown in Fig. 2.

3) Federated Learning Model: It is a kind of distributed model but with the facilitation of global knowledge collected from all the distributed clients. Same as a distributed setting, a general or pretrained model is distributed to clients initially.

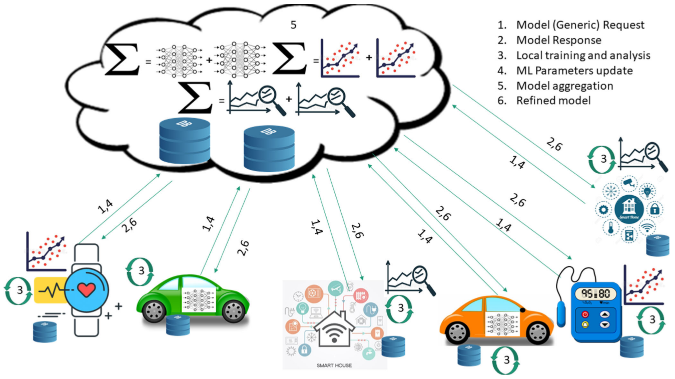

Fig. 3. FL model for IoT.

All the clients personalize the model locally with its local raw data. Clients perform ML tasks locally and send their parameters to the server. The server then aggregates all the updates received from the clients and performs ML tasks and finally, distributes the updated model to the clients [11]. This is an ongoing process by which the clients are constantly provided with all the new and emerging global knowledge. The working model of FL is shown in Fig. 3. This learning model first formulated by [21] is as follows:

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \text{ where } F_k(w) = \frac{1}{n_k} \sum_{i \in P_k} f_i(w). \quad (1)$$

In (1), $f_i(w)$ represents a loss function of prediction for input x_i to an expected output y_i with weight vectors w. K is the number of participants in the current learning round and $F_k(w)$ is the local objective function of the kth participant. For the total number of samples n, n_k is the number of samples present locally in the kth participant. Similarly, P_k with $n_k = |P_k|$ is the partitioned assigned to the kth participant from the whole data set P.

In a typical FL setting, when a device downloads the current model parameters (weight) from the server first, it initializes the local model with the downloaded parameters, and then the local data set is used to train the model. The parameters are optimized by minimizing the local objective function that uses stochastic gradient descent (SGD). The optimized parameters from all such devices are sent to the server where they are aggregated using the FederatedAveraging algorithm [21]. This way, the global model is updated and the learning takes place.

As raw data reside locally on the device and only ML parameters are sent to the server, FL ensures privacy of the raw data of clients and complies with privacy policies and/or regulations, e.g., The European Data Protection Regulation general data protection regulation (GDPR) [22]. Additionally, FL frameworks are also enriched with privacy-preserving techniques, such as differential privacy [23], secure multiparty computation (SMC) [24], and homographic encryption (HE) [25], to send the ML parameters from clients to the server securely. Despite presenting propitious potential, FL brings several challenges when it is applied with IoT. Here, we highlight some major challenges associated with FL for IoT.

1) *Limited Device Memory:* IoT devices constantly generate data during their operation. Due to their limited memory, when the batch size of data increases, training the federated model locally is not feasible. In an FL scenario, these devices might be dropped out or are forced to use a simple model to work with small batch sizes in the training phase [1].
2) *Limited Battery Power:* If the learning model is complex and the training data size is huge, IoT devices might be run out of battery power during the training phase.
3) *Limited Computing Power:* IoT devices, in particular, are limited to computing power. Due to this constraint, training the model locally by such devices may not be a feasible approach.
4) *Vulnerability:* We have seen an unprecedented growth of diverse sets of IoT devices in recent times. Some categories of IoT devices are vulnerable enough to be gain controlled by hackers. Such devices might produce malicious data and when such data are used to train the model, it might even affect the global or federated model.
5) *Unreliable and Limited Availability:* In FL, clients can drop out anytime. Clients might be dropped out by several factors, such as unreliable network connection, limited storage, computation power, and more. Moreover, the availability of clients depends on time and location. More clients might be available during day time compared to night time. Day and night time also differ by geographical location.
6) *Stateless:* The availability of clients depends on several factors and so the client does not guarantee repeated computation.
7) *Anonymity and Poisoning:* Clients in FL are anonymous, which makes it hard to differentiate between genuine or malicious clients. So, there might be a chance that the federated model might get poisoned by the involvement of malicious clients.
8) *Nonindependent and Nonidentically Distributed (Non-IID) Data:* The nature of local data on a device depends on its unique behavior and usage pattern and so the distribution of clients and data is nonuniform. The data of the same device might differ because of the change in location, time, and users.
9) *Local Training:* Each client is limited to its local data. Nonenough data on a device might not be able to train and produce a good model.
10) *Accuracy:* Due to the characteristics of FL, such as Non-IID data, stateless, local training, and resource constraint, the aggregated global model might not be as accurate as compared to centralized learning. The nonaccurate global model in turn might affect the local model and as the chain reaction, the global model is again getting more affected.
11) *Communication Overhead:* The frequency of communication for a client with a server not only depends on factors, such as its characteristics, size, and quality of local data but also might be heavily influenced by other clients. Frequent communication with servers to keep the

local model consistent with the global model increases communication overhead.

Since the first proposal of FL in [21], there have been several research to address challenges that exist in FL. For example, to reduce the communication overhead by aggregating global model only when the global model's weight differs by some empirically selected threshold is proposed [26]. For a similar issue, a control algorithm to find global aggregation frequency was proposed in [27]. To mitigate the effect of non-IID data and improve the accuracy, a feature fusion approach by aggregating local and global model is presented [28]. To address a similar issue, Sattler *et al.* [29] designed a federated multitask learning (FMTL) framework to form clusters of clients based on the geometric properties of the FL surface with jointly trainable data distribution. Work in [20] uses deep reinforcement learning agents based data offloading decisions to address resource constraint issues and other challenges and make FL operations efficient. Detecting sybil-based parameter poisoning from the diversity of client updates in the distributed learning process and taking corrective measures is proposed in [30]. Several works [31]–[33] have proposed IDSs in FL setting that learn from global knowledge of threats and detect new and emerging cyberthreats. We discuss several recent works that address challenges and issues that exist in FL in Section III.

B. Typical Types of Cybersecurity Models

Security is the fundamental requirement of today's digital world. An exponential rise of vulnerable heterogeneous IoT devices and furthermore, communicating through a wireless medium, has widened the attack surface significantly. Wireless communication networks' standards and protocols are different but more vulnerable than wired communication networks. The mobile and distributed nature of the IoT devices exaggerates the security challenges even more. So, the security solutions designed for wired networks cannot be directly applied to the wireless network. Similar to learning models, cybersecurity models for IoT environments can be categorized into three types as isolated devices level cybersecurity model, distributed cybersecurity model, and FC model (as shown in 4). We can think of these as cybersecurity models that provide security services working at different levels. Adopting one specific type of the security model is insufficient, so an effective cyberdefence mechanism is likely to require the combination of such models working in place.

1) Isolated Devices-Level Cybersecurity Model: This cybersecurity model works at the lowest level and concerns with providing security services to the end devices. Due to the heterogeneous nature of IoT devices, each category of devices might have specific vulnerabilities and security requirements. So, the device-level cybersecurity model needs to take care of safeguarding the device against any malicious activity. From the basic security measures, such as password setting, validating authentication, and access control, it aims to validate each connection request and establish secure communication to the outside world. Device-level security also aims to validate the timely software updates and makes sure the update process is completely secure. Furthermore, it also aims to safeguard the device against malware attacks. Although device-level cybersecurity model intends to provide all the essential security measures, it is not sufficient to fully protect the system. Attackers use sophisticated tools and codes to generate new and polymorphic malware to attack the connected system. So, it necessitates the device-level security to be backed by ML models to learn and adapt based on the dynamic scenarios. It should be capable of taking defense mechanisms on any attacked or anomalous situations and allowing smooth device operations. However, most of the IoT devices are resource constraint, which makes them incapable of running ML models. To deal with it, in an IoT network, a gateway node or edge node is typically employed for running ML-backed cybersecurity model and providing necessary security to all the end devices connected in the network.

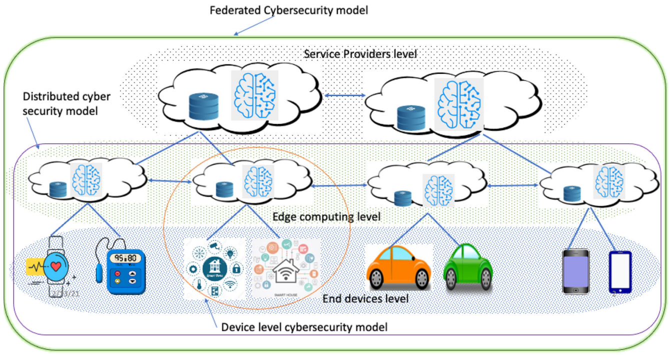
Federated Cybersecurity model
Service Providers level
Distributed cyber security model
Edge computing level
95.80
End devices level
Device level cybersecurity model

Fig. 4. FC models for FL in the IoT.

2) Distributed Cybersecurity Model: A significant number of new cyber threats are being introduced every day. Learning from cyber attacks/threats from one IoT network is not sufficient. In a distributed network, edge nodes are geographically dispersed and are closest to the end devices or users. So, a distributed cybersecurity model aims to enable collaboration and cooperation among geographically distributed edge nodes to provide better security services. Based on the characteristics of the underlying IoT network, edge nodes among themselves may be distinctive for the specific security services they offered. If any edge node cannot provide the intended service to a nearby device or user, it collaborates with other nodes at the same level to do so. Such collaboration facilitates to provide appropriate security solutions to combat emerging cyber threats/attacks in the real-time scenario.

3) Federated Cybersecurity Model: It is a cybersecurity model as shown in Fig. 4, that provides security and other services from the top level in the federated model based on the feedback from the bottom/device level (e.g., [34] and [35]). IoT service providers participate in this level to provide the necessary services to their respective users or devices. Each user can access the respective services from its service provider. The edge node on a particular IoT network acts in the middle to ensure the necessary security and services are provided to its end users or devices. Each service provider is responsible to disseminate essential security services to all its distributed devices through edge nodes. In this security model, each service provider learns from all its devices and updates the security model accordingly. Furthermore, these independent service providers also collaborate themselves to make dynamic defense strategies/solutions to combat against possible attacks/threats. In the immediate lower level, if edge collaboration could not provide a security solution in real time, a particular edge node reaches out to its service provider. The service provider then provides the necessary security solution or collaborates with other providers to do so.

4) Federated Learning and Federated Cybersecurity: The existing approach of the FC model provides security solutions to IoT applications through communicating and collaborating at different levels as needed. However, the traditional way of exchanging data/information within the same level and/or between different levels can pose privacy and security concerns. FL has been emerged as a solution to exchange data/information in a secure and privacy-preserving way. A FC model accompanying FL to collaborate and exchange any information at any level offers a huge potential to make the IoT network safe and secure. Most of the FC approach utilizing FL as a cyber-defense mechanism primarily focused on securing IoT networks considering a single global model offered by a single service provider. However, this approach can easily be extended to a collaborative scenario involving multiple global models maintained by different service providers. Only a few research have worked toward creating a sense of federated security model utilizing multiple global models. We present a survey of several research efforts toward creating FC models for IoT network using FL in the next section.

III. Recent Advances on Federated Learning for Cybersecurity and Cybersecurity for Federated Learning

The focus of this work is to survey several existing works since 2015 toward cybersecurity particularly for IoT environments. The addressed issues by those works and the environments where they are implemented or tested are given in Table II. In recent times, a significant number of research works for addressing security in the IoT networks have been shifted toward applying FL. The framework of FL inherently supports privacy, to some extent security, and latency as only updates are required to transmit but these are costlier to achieve in centralized learning. Distributed learning addresses these issues but lacks global knowledge of collaborative learning. There are some downsides of FL in IoT networks too, such as the heterogeneity of devices, resource constraint, non-IID data, accuracy, and others. Mainly, most of the FL surveyed works address security and privacy issues but there are several works that also address issues, such as latency [26], [36]–[41], resource constraint [20], [27], [42]–[45], accuracy [28], [41], [46], and non-IID [28], [29], [39]. All these issues are somehow dependent on each other and improving one issue should not affect the others. Some works have considered all these issues while others addressed the only subset of these. We will discuss some of the contributions made to alleviate such issues present in FL. Although FL in the IoT environment is our primary focus of study, some recent works we studied are proposed and tested in the distributed learning setting. We have also mentioned those works considering their usefulness to secure IoT environment and are easily extensible to FL setting.

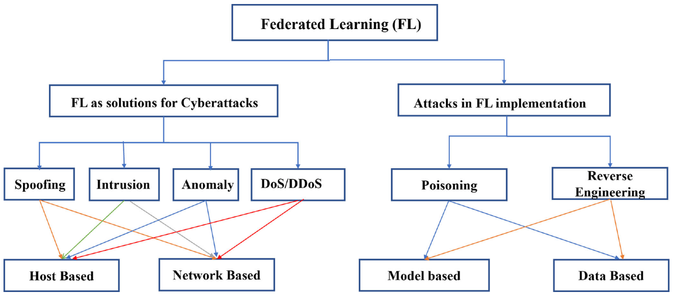

Fig. 5. FL as a security solution to different attacks and novel attacks present in FL.

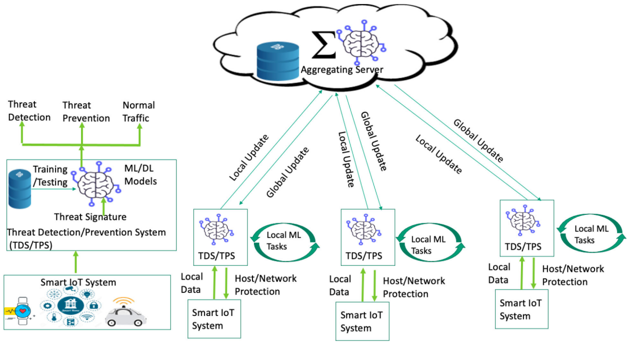

Fig. 6. FL as a solution to mitigate possible threats in the IoT network.

We have summarized surveyed works into two groups. In one group, we discuss existing works related to FL as a tool for cybersecurity and in the next, we present works based on cybersecurity need for FL. FL as a solution to different types of attacks and FL as a target to different potential cyberattacks are highlighted in Fig. 5. A collaborative approach of identifying and learning different types of attacks can be highly effective to mitigate daunting threats, such as intrusion, Dos/DDos, anomaly, and others. On the other hand, before utilizing FL for real applications, the emerging attacks typical to FL are required to be addressed.

A. Federated Learning for Cybersecurity

Security, privacy, and trust have been extensively studied in the literature in the context of cyberspace. However, this survey is particularly focused on cybersecurity for IoT environments in the FL setting. The IoT environment is more vulnerable to different types of cyberattacks, so a collaborative learning framework of FL only by sharing the model update can be an effective solution to enhance security and privacy. A timely learned and shared global knowledge of different types of cyberattacks, such as spoofing, intrusion, anomaly, and DoS/DDoS, facilitates building and enhancing cyberdefence models and mechanisms accordingly. So, FL has a huge potential to secure cyberspace effectively both in the device as well as network level. The application of FL as a solution to mitigate possible threats is depicted in Fig. 6.

In recent times, cyberspace has been more vulnerable due to the presence of unprecedented growth of heterogeneous sensor devices. IDS and anomaly detector backed by ML has become mandatory to detect and combat intrusions and

TABLE II
ADDRESSED ISSUES BY SURVEYED WORKS

Addressed Issues	References	FL	Domain
Security,IDS, IPS	[47]	✗	Smart home
Security, DDoS Resiliency	[48]	✗	IoTs environment
Security, IDS	[3]	✓	IoT network
Malware classification	[49]	✓	Edge devices
Security, IDS	[50]	✓	Network environment
Security	[51]	✗	IoTs environment
Security	[52]	✗	IoTs environment
Security, IDS dataset	[53]	✗	IoT and IIoT
Security,IDS dataset	[54]	✗	-
Security	[55]	✗	IoTs environment
Security, IDS	[56]	✗	VANET
Security, IDS	[57]	✗	Network
Security, IDS	[58]	✗	Network environment
Privacy, Security, IDS	[32]	✓	CPSs
Security, ADS	[59]	✓	SOHO IoTs
Security, ADS	[33]	✗	Smart city IoT
Cyberattacks	[60]	✗	IoTs, CPSs
Cognitive cybersecurity	[61]	✗	CPS-IoT Enabled Healthcare
Privacy, integrity	[62]	✓	Edge devices
Security, Sybil based poisoning attack	[30]	✓	Edge network
IoT Mirai botnet attack	[63]	✗	IoTs devices
Reliability, Security	[64]	✗	IoTs network
Security, Audit	[65]	✓	Edge network
Security, Trust	[66]	✗	IoTs network
Security	[67]	✗	IoTs network
FL operation, Security	[68]	✓	Overall FL framework (IoTs, Edge cloud, Regional Cloud, Core Cloud)
Privacy, Security, Latency	[36]	✓	IoT edge computing (Connected vehicles)
Jamming attack detection and defense	[69]	✓	UAV
Security, Privacy Throughput, Latency	[37]	✗	IoT network
Privacy, Security, Communication overhead, computational cost	[38]	✗	Fog-based IoT
Gradient sparsification, Accuracy	[70]	✓	IoT edge computing
Security, Intrusion, Privacy, IDS	[31]	✓	IoT devices
Privacy, IDS	[71]	✓	Edge devices
Privacy, Latency, Non-iid	[39]	✓	IoT network
Latency	[40]	✓	IoT network
Learning speed, Accuracy	[46]	✓	Edge devices
Increased accuracy, Convergence process	[28]	✓	Edge devices
Communication, Accuracy	[41]	✓	IoT environment
Efficient communication and training	[26]	✓	IoT environment
Resource constraint, Global aggregation frequency	[27]	✓	IoT environment
Security, Resource constraint	[42]	✗	IoTs environment
Resource constraint	[20]	✓	IoT edge computing
Resource constraint	[43]	✓	WAN
Resource constraint	[44]	✓	IoT edge computing
Privacy, Latency	[72]	✓	Edge network
Non-iid, Accuracy	[29]	✓	Edge network
Resource demand, Scarcity of relevant data, Security, Latency	[73]	✓	IoT Edge network
Privacy, Security	[74]	✓	IIoT
Privacy, Security, IDS, Accuracy	[75]	✓	IoT environment
Security, Data collaboration	[76]	✓	IoT environment
Privacy, Security, Reliability	[77]	✓	IIoT environment
Safety, Resiliency Accuracy, Privacy, Latency	[78]	✓	IIoT environment
Security, IDS, Communication	[79]	✓	IIoT environment

anomalies in today's gigantic cyberspace. In the literature, different approaches (e.g., [32], [58], and [59]) using varieties of ML algorithms [e.g., convolutional neural network (CNN), nonlinear autoregressive (NAR), and Q-learning] have been examined to design IDSsIPSs and those are tested against several benchmarked data sets for its performance. Majority of the efforts dedicated to designing FL-based security solutions primarily focused on the accuracy of the security model only without considering other important performance metrics. We cover FL works addressing performance issues in the next section.

Rahman *et al.* [31] proposed an FL-based self-learning IDS to secure the IoT environment. A benchmarked data set (NSL-KDD) consisting of normal traffic and several attack types was first distributed over the IoT devices and then the ML-based IDS model was trained and tested locally. The model updates were sent and aggregated following the conventional FL operations. The proposed system achieved accuracy close to the centralized learning approach. The FL approach was successful to create a self-learning IDS by which end devices were successful to detect attacks that were not presented in their local data set. The advantage of such FL-based IDS is that in a real application scenario, IDS can be capable to detect intrusions not generated previously by its own traffic. The downside of the proposed approach is that it was experimented within a significantly small IoT network environment and except accuracy, other performance metrics were not considered.

In [78], a collaborative IDS (CIDS) is developed as smart "filters" by deploying at IoT gateways in each subnetwork. DNN of each filter is trained with a local database housed in subnetwork and such learned models from the filters are collected and aggregated in a central server. Each filter supplemented by global knowledge is capable of detecting and preventing real-time cyberattacks. The performance of the proposed model was tested with multiple benchmarked data sets and it outperformed several baseline ML models in FL and centralized learning settings in terms of detection accuracy, network traffic, privacy, and learning speed. Despite the improved performances in several aspects, this approach is useful against known attacks only.

A robust FL-based IDS using a generative model was envisioned in [79]. FED-IIoT, an FL-based architecture for detecting malwares used generative adversarial network (GAN) and federated GAN (FedGAN) algorithms in the participant side to generate adversarial data and inject them into the data set of each IIoT application. On the server side, a robust collaboration of trained models was ensured by incorporating a defense mechanism to detect and avoid anomalies while aggregation. The proposed model demonstrated higher accuracy compared to existing solutions and allows secure participation and efficient communication among participants in the IIoT environment.

With similar objective, work [32] designed an ML-based IDS model to detect threats in the industrial CPSs environment. The designed IDS model was further extended as an FL framework to allow multiple industrial CPSs collaborate to build a comprehensive IDS. The authors compared the effectiveness of the proposed model with state-of-the-art schemes through extensive experiments on a real industrial CPS data set. For ensuring security and privacy of the federated model parameters, the authors incorporated the Paillier cryptosystem-based secure communication protocol for the federated IDS. The advantage of this work is that it makes FL secure against the man-in-the-middle type attacks.

Aiming to identify the most critical cyberattacks in a smart home environment, [47] first highlights attack surfaces and prepares three test cases (to test confidentiality, authentication, and access control) to launch different types of cyber security-based attacks. An IPS is then designed and tested against the same attacks to verify the resiliency of the affected system.

In an effort to detect cyberattacks in a larger IoT network, an ML-based network IDS (NIDS) capable of monitoring all the IoT traffic of a smart city in a distributed fog layer was proposed in [33]. The proposed model performed well to detect attacked IoT devices at distributed fog nodes and alert the administrator accordingly. The NIDS model was evaluated against the UNSW-NB15 data set [80] and the model demonstrated the classification accuracy of 99.34%. The authors claimed their approach as unique stating that the NIDS model learns with normal traffic and can detect malicious behavior in the future.

Extending the traditional FL model, Sun *et al.* [3] proposed a segmented FL framework to detect intrusion for large-scale networked LANs. This approach is different from a traditional FL model that works on collaborative learning based on a single global model. The proposed approach instead keeps multiple global models where each segment of participants performs collaborative learning separately and also rearranges the segmentation of participants dynamically. Moreover, these models interact with each other to update parameters as per the various participants' LANs. The authors employed three types of knowledge-based methods for labeling network events and training a CNN using a data set. The model was trained and tested using a data set consisting of using two months' traffic data set of 20 participants' LANs and obtained a high validation accuracies. The advantage of the segmented FL framework is that it performed better to detect intrusion in LANs compared to the traditional FL approach of using a single global model.

A CIDS to detect abnormal network behavior in the whole VANET was proposed in [56]. The CIDS used DL and SDN controller approach to train a global IDS that can work in both IID and non-IID situations. Instead of directly exchanging subnetwork flows, multiple SDN controllers were employed to train global IDS jointly for the entire network. The model was built and tested using KDD99 and NSL-KDD data sets to validate the efficiency and effectiveness of the CIDS for VANETS. The main highlighting feature of the proposed approach is that the CIDS is effective to detect intrusion in the entire VANET and not just limited to the local subnetworks like other approaches.

To alleviate Wi-Fi network privacy concerns, a federated DL model [71] was built and tested using the Aegean Wi-Fi intrusion data set (AWID). The proposed model used a specialized DL neural network called stacked autoencoders (SAEs) to capture a compressed representation of anomalous observations. To identify the new threats, the federated model learns from the new observations and updates the local and global models. The result obtained was compared with the classical DL model and claimed that the FL model was more effective in terms of classification accuracy, computation cost, and communication cost. This work is different than others to use a specialized DNN, which facilitates compression of model parameters that mainly benefits to reduce communication latency.

To deal with the emerging sophisticated polymorphic threats, a security solution needs to be proactive to identify unforeseen and unpredictable cyberattacks. In an attempt to design such a solution, Rege *et al.* [58] extended IDS to offer temporal prediction of adversarial movement. The proposed approach used four predictive models, namely, NAR neural network, NAR neural network with exogenous input

(NARX), NAR neural network for multisteps-ahead prediction, and autoregressive integrated moving average (ARIMA), and compared the results over two data set collected at different locations. The research was able to identify five advanced persistent threats' trends—there will be more attacks, more obfuscation, continued false attribution, greater shifts from opportunity-based attacks to more targeted attacks, and more damage ranging from data manipulation to data encryption or deletion.

Motivated by the similar need, Vinayakumar *et al.* [57] presented several experimental approaches to identify the best algorithm to design dynamic IDS that could effectively detect and predict intrusions at both host level and network level. The authors first experimented with various DNNs against publicly available benchmark malware data set (KDDCup 99) by choosing optimal network parameters and network topology for DNNs. The well performed DNNs are then tested with other malware data sets NSL-KDD, UNSW-NB15, Kyoto, WSN-DS, and CICIDS 2017 to set the benchmark. A similar approach was followed to identify well-performed classical ML classifiers and to compare its performance with DNNs. The performance evaluation demonstrated that DNNs outperformed classical ML classifiers and finally, the authors utilized the better performed DNNs to design a highly scalable and hybrid DNNs framework called scale-hybrid-IDS-AlertNet. The proposed IDS could not only monitor real-time network traffic and host-level events effectively but also proactively alert possible cyberattacks.

A federated self-learning anomaly detection and prevention system that is capable of detecting and preventing emerging and unknown attacks in the IoT network (DÏoT) was proposed in article [59]. Without human intervention, DÏoT builds device-type-specific communication profiles that are eventually used to detect anomalies in devices' communication behavior. Security gateways were employed in such a way that each gateway is assigned to monitor the traffic of one particular device type. The collected traffic data were then used to train the local model of each gateway and the model parameters of the training were sent to an IoT security service for aggregation. IoT security service had been used as a repository of device-type-specific anomaly detection models, which in the later stage also used to aggregate all the updates received from security gateways.

Pang *et al.* [50] proposed a learning agent-based federated network traffic analysis engine (FNTAE) for detecting real-time network intrusion. To detect abnormal traffics as a result of new attacks, the proposed model made use of an analysis engine powered with an incremental learning agent to capture attack signatures in real time. FNTAE demonstrated well compared to the centralized analysis system, however, it is useful only to combat against the known attacks.

To secure an IoT environment, some works have followed other approaches too. The work presented in [51] proposed man-in-the-middle-IoT-computing tool (MIMIC), which utilizes the man-in-the-middle attack concept to deploy MIMIC as a fog computing agent for IoT networks. MIMIC is deployed at the edge node of the IoT network to be able to sniff, capture, and replay all the incoming packets from IoT devices. MIMIC then creates a virtual layer for holding the virtualization of all the sensing devices and the remote users are allowed to query only on the virtual space disabling the direct access to physical devices. Zarca *et al.* [52] proposed a novel approach of utilizing SDN and NFV to deploy IoT honeynets to distract cyberattackers and make the IoT system secure. Administrators of the IoT system can deploy IoT honeynets as a service through high-level security policies defined over SDN controller and NFV Management and Network Orchestration by replicating the physical IoT architecture on a virtual environment as VNFs. The model experimented in a testbed of H2020 EU project premises and it was successful for filtering, dropping, and diverting the network traffic dynamically, and adapting the network behavior according to the new deployed vIoTHoneyNets (virtual IoT honeynet) needs.

There have been other significant research to study cyberattacks and build corresponding cyberdefense mechanisms that use different approaches, utilizing varieties of databases, API, platforms, frameworks, and ML algorithms. For example, a malware classification prototype accompanied by decentralized data collection and sharing using the FL model approach was developed in [49]. The data set of 10 907 malwares obtained from virustotal api was used for training and testing the model. The authors used SVM and LSTM ML algorithms in a federated setting to achieve better results on the classification of malwares. A framework called DRAFT is developed in [48] by integrating other frameworks and tools to improve the resiliency of the end-to-end IoT platform against cyberattacks. The proposed model was integrated in the IoT platform and tested against five known simulated cyberattacks using Fed4FIRE+ federated testbeds and demonstrated the increase in cyberattack resiliency for the tested IoT platform. An adaptive federated reinforcement learning was proposed in [69] to combat jamming attack in unmanned aerial vehicles (UAVs). The proposed model used model-free Q-learning and CRAWDAD data set and learned jamming defense strategy in a newly explored environment. Paper [60] studies cybersecurity in the context of Big Data IoT and CPS. Cybersecurity issues and vulnerabilities associated with CPS were investigated and analyzed to pinpoint possible cyberattacks. The authors also presented technical approaches to mitigate those attacks. Abie [61] proposed a four-layer architecture of cognitive cybersecurity to combat against dynamic and adaptive attacks in smart CPS-IoT enabled healthcare environments. The presented conceptual architecture aimed to mimic the cognition behavior of humans to anticipate and respond to new and emerging cyber threats in the smart healthcare domain. In another work of providing cybersecurity for IoT devices [42], the authors presented an approach of incorporating a trusted network edge device (NED) developed in [81] as a proxy service for IoT communication. To protect IoT devices, users can set up security solutions and policies easily and efficiently for multiple IoT gateways and end devices at once via NED. The proposed approach is experimented in a corporate scenario in VTT Oulu premises. A work presented in [67] highlights several hardware-assisted techniques employed in the literature that can be applied to add another layer of protection to combat cyberattacks in the IoT domain. The paper also explored the hardware solutions with respect to cost, performance, security, and presented challenges to adopt in real scenarios.

To improve security and reliability in an IoT environment, a reliable and efficient adaptation of cluster techniques (REACT) was presented in [64]. In REACT, an effective cluster head selection algorithm and energy balanced routing algorithm were proposed and simulated with estimated parameters against existing protocols HEED and LEACH comparing throughput, network lifetime, energy remaining, and reliability. The paper also presented a strategy of a cyber-hacking technique of selecting an attack point to improve the cybersecurity design. With the aim of facilitating the design of an effective IDS and evaluating it properly, some works have dedicated efforts to fill the gap of the availability of benchmarked intrusion data set to test IDSs-enabled IoT systems. The work presented in [53] proposed a new data-driven IoT/IIoT (TON_IoT) data set containing Telemetry data of IoT/IIoT services, Operating Systems logs, and Network traffic of IoT network, collected from a realistic representation of a medium-scale network at the Cyber Range and IoT Labs at the UNSW Canberra (Australia). TON_IoT also contains label and type features indicating multiple classes and subclasses suited for IoT/IIoT applications for multiclassification problems. The features of the data set were compared with other existing data sets to show its superiority. In another example, Sharafaldin *et al.* [54] produced one of the most popular intrusion data set named CICIDS2017, which contains an important set of features and meets real-world criteria. The produced data set is fully labeled containing more than 80 network traffic features and meets all the required criteria with common updated attacks, such as Denial of Service (DoS), DDoS, Brute Force, XSS, SQL Injection, Infiltration, Port scan, and Botnet.

In this section, we discussed several existing approaches to design cybersecurity models particularly for IoT environments and in FL setting. Many ML algorithms, blockchain, network virtualization, SDN, clustering approaches, and others have been explored aiming to design an efficient cyber defense mechanism to detect and prevent intrusion, anomaly, Dos/DDoS, and other attacks in different types and sizes of IoT networks.

B. Cybersecurity for Federated Learning

We presented several works discussing FL as an effective solution for different kinds of security and privacy issues. However, this new learning architecture has invited some novel kinds of attacks. In the FL setting, although the data reside locally in end devices and only ML parameters are exchanged between client and servers, it is still vulnerable to different kinds of attacks. We first discuss different types of attacks to FL and then present the mitigating strategies proposed in the research.

Parameter poisoning (or model poisoning) and reverse engineering ML attacks are some serious threats in FL and are an active area of research (e.g., [30], [82]–[84], and [74]). The typical attacks in FL can be data based or model-based (as shown in Fig. 5), which can be performed by forging local data of end device(s) or the model parameters on client or server side. How an attacker may perform different attacks in FL is shown in Fig. 7. As depicted, an attacker may control IoT device/network to compromise local data and/or local ML tasks to generate poisoned model. In other scenarios, an attacker may perform man-in-the-middle attack to forge the model update in transit or just to overhear communication to reveal the privacy of a user.

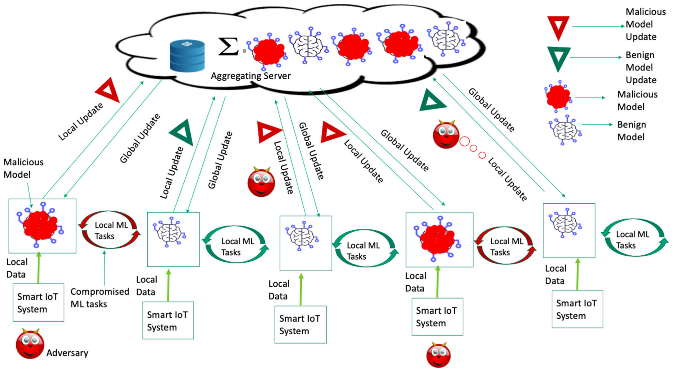

Fig. 7. Attack scenarios in FL.

The attacks in FL can not only degrade the quality of the learning model but also expose the privacy of users. An adversary can reveal the privacy of a user by spoofing on model updates sent by the user's device. Moreover, if the adversary gains control of the aggregating server, he/she can get comprehensive knowledge of the history of update parameters of devices and the structure of the global model. With these information, adversaries can reveal the privacy of devices through reverse engineering.

With access to the model updates, some works demonstrated generating pictures that look similar to the training images using GAN (e.g., [85] and [86]). Extending the leakage of private information to the next level, Zhu and Han [87] demonstrated that an attacker can completely steal the private training data from the shared model updates in a few iterations. To achieve this, the authors first generated a pair of dummy inputs and labels and which were used to generate dummy gradient following the common training process. Rather than optimizing weights, they optimized dummy inputs and labels so as to minimize the distance between dummy gradients and real gradients and were successful to reveal the training data completely. Furthermore, with the full control of central server, adversaries might forge the global model, which, in turn, might affect the local model of the end devices. In effect, aggregating local updates of such models might degrade the quality of the global model significantly. Even if adversaries do not have control over end device or server, the model parameters might still be forged while in transit between the client and the server.

On the other side, FL is also vulnerable to data poisoning and model poisoning attacks performed through end device(s). If an adversary gains control over an end device, he/she may forge the local data and/or forge the model update during the local model training process with the intention of creating a biased model. The parameters of the biased model, in turn, might affect the quality of the global model. This problem gets even worse in case of the Byzantine problem [88] and Sybil attack [89]. A survey presented in [14] categorizes and discusses threats to FL and presents future research directions to create robust FL framework.

The label flipping attack is one of the most common data poisoning attack where the labels of training examples of one class are changed to another class (keeping features of the examples unchanged) to force the model predict incorrect label. Fung *et al.* [30] demonstrated the label flipping attack by flipping the label 1 s in the training data set to label 7 s and making the model incorrectly classify 1 s as 7 s. In other form of the data poisoning attack, an attacker may change individual features of the original training data set to plant backdoors into the model [14]. The general approach behind the backdoor attack is to replace the global model with the attacker's model and force it to mispredict on a specific subtask, e.g., compelling an image classifier to misclasify green cars as frogs [90]. Once the estimate of global model's state is perceived, an attacker can replace the model with a simple weight rescaling operation [91]. Bagdasaryan *et al.* [90] exhibited the backdoor attack by injecting certain pattern to the data and altering the label to the desired target so as to mislead the global model. The attack scenario consisted of one or more malicious participants, which train on the backdoor data and then share the model update to the server for aggregation.

Data poisoning ultimately poisons the model update, however, an attacker may directly manipulate the training process without poisoning training data and it is to be noted that this form of model poisoning is regarded as more effective than data poisoning. Bhagoji *et al.* [82] demonstrated using model poisoning attacks considering a single, noncolluding malicious agent with the adversarial objective of causing the FL model to misclassify a set of chosen inputs with high confidence. To make the targeted misclassification effective, the authors employed malicious agent's update boosting as well as the alternating minimization strategy to alternately optimize the training loss and the adversarial objective. In another example, Blanchard *et al.* [92] exhibited model poisoning considering the omniscient attack (adversaries with aware of good estimate of gradient) where adversaries send opposite update vector by multiplying with negative constant to reverse the direction of gradient descent and degrade the model performance. Furthermore, Baruch *et al.* [93] demonstrated that model poisoning through the Byzantine attack is still possible in a nonomniscient attack scenario by introducing even a small but well crafted changes on gradient.

Byzantine-tolerant learning in the distributed setting has been addressed in some works (e.g., [86] and [94]–[97]) where most of them assume participant's data are i.i.d, unmodified, and equally distributed. However, in FL, data distribution is different and there solutions are not fully applicable. Bagdasaryan *et al.* [90] exploited the solutions presented in [86], [95], and [97] and was able to partially mitigate the attack but that is also at the cost of global model's accuracy. To address model poisoning, Fung *et al.* [30] first demonstrates the FL's vulnerability against the Sybil-based poisoning attack through experiment and presented an FL model FoolsGold that identifies such attack based on the diversity of client updates in the distributed learning process. This model even works effectively in case Sybils compromised honest users. The advantages of this system compared to prior approaches are that it is not bounded by the expected number of attackers, it does not require extra information outside of the learning process, and it works with fewer assumptions about clients and their data. However, combating against a single client adversary and improving the model against informed attack are some limitations of this model.

Blanchard *et al.* [92] first confirmed that federated averaging (FedAvg) does not resist Byzantine attacks and then proposed a Byzantine-tolerant aggregation rule called *krum* to address the model poisoning attack. Considering f byzantine attackers out of n participants in a communication round, *krum* first calculates the pairwise euclidean distance of n-f-2 updates that are closest to a model update δ_i and then computes the sum of squared distances between δ_i and its closest n-f-2 updates. Finally, the algorithm updates the global parameter by the model update with the lowest sum. The idea behind this is to choose a vector that is somehow the closest to n f workers and guarantee convergence regardless of f Byzantine attackers.

A work presented in [98] proposed an aggregation rule considering no bound on the number of Byzantine workers but still demonstrated better convergence. The proposed approach computes a score for each worker using a stochastic first-order oracle to determine its trustworthiness. The server ranks each candidate gradient estimator as per the estimated descent of the loss function and the magnitudes. It then calculates the averaged gradient over the several candidates with the highest score. The server compares the true value of the gradient with the average gradient to identify whether the update is harmful or not.

Sun *et al.* [99] studied the vulnerability of FL for data poisoning and devised a bilevel optimization framework adaptive to the arbitrary choice of target nodes and source attacking nodes to compute optimal poisoning attacks. Exploiting the data collection process, an attacker can directly inject poisoned data to all the target nodes. The authors also considered an indirect way of poisoning data to target nodes by exploiting the communication protocol in case direct attack is not possible. This work highlights challenges associated with FL where attackers can exploit the communication protocol to open a backdoor to lunch data poisoning attacks. To adopt FL as a probable cybersecurity solution, a cybersecurity mechanism to combat possible threats in FL should be in place. So, we also discuss some research works that present the cybersecurity solutions to the potential threats existed in FL.

To address backdoor attacks in [100], the authors presented defense approaches using norm clipping and differential privacy. Norm clipping was considered to combat boosted attacks, which are likely to generate updates with large norms. This approach was used to put a bound on the sensitivity of the gradient update by ignoring updates if its norm is above some threshold norm. Furthermore, the authors also used differential privacy to supplement norm clipping by adding Gaussian noise to the updates to mitigate the effects of adversaries beyond norm clipping.

In FL, if an attacker does not have a control over the clients, it is still quite possible to lunch man-in-the-middle attacks. He/she can overhear model updates to reveal the privacy of clients and even can forge model updates in transit. To address this attack scenario, techniques, such as differential privacy [77], homomorphic encryption [101], [102], secure function evaluation or multiparty computation [103], and other cryptographic approaches, have also been applied on top of FL. Differential privacy is effective to preserve privacy of clients due to added noise on shared model updates and thus, mitigates reverse engineering attack while other approaches even mitigate any chance of manipulation of model updates while in transit. However, these approaches adds up computation and communication burden compared to the differential privacy approach. Geyer *et al.* [104] proposed an algorithm for client-sided differential privacy-preserving federated optimization. It is demonstrated that client's participation can be hidden at the cost of minor loss in model performance when sufficient client participates. Zhang *et al.* [105] also used a differential privacy approach to protect patients' privacy against possible reverse engineering attack.

Zhu and Han [87] first demonstrated reverse engineering attacks and then presented some defense strategies. Approaches, such as adding noise on gradients before sharing, gradient compression and sparsification, and others, were experimented to observe its performance against information leakage. To address reverse engineering attacks by preserving the privacy of end users, Al-Marri *et al.* [75] adopted the mimic learning approach [106] to work in the FL scenario. Mimic learning used two kinds of learning models named as a student and a teacher. The student model is trained with a public data set whereas the teacher model is trained with sensitive user data. Then, the teacher model is used to label the public data set, which is later used to create a student model and sent to the centralized server for generating a new global model. The approach of transferring knowledge from the teacher model to the student model without revealing any sensitive information was used to protect the student model against reverse engineering attacks.

To strengthen privacy by securing the parameters exchange between client and aggregating server, homomorphic encryption[1] is one of the techniques in which aggregation can be performed directly on the encrypted parameters. This approach allows aggregation without revealing model updates, which secures FL from any kind of spoofing or manipulation of model updates. Taking the computation and communication overhead of this approach into account, Zhang *et al.* [101] proposed an efficient homomorphic solution called BatchCrypt. To apply this solution, first, new quantization and encoding schemes together with a gradient clipping technique were developed. After this, instead of applying homomorphic encryption on individual gradients, BatchCrypt was used to encrypt an encoded batch of quantized gradients. BatchCrypt demonstrated significant speedup in training and reduction in communication overhead (compared to encrypting each gradient) with negligible loss in accuracy.

Moreover, in recent times, blockchain technology (BC[2] has been extensively applied for many applications due to its decentralized, auditable, secure, and privacy-preserving features. So, some research works (e.g., [76] and [77]) have incorporated blockchain in FL setting too. To mitigate the effect of revealing sensitive information while sharing gradient and chance of forging aggregated gradients by a malicious server, a verifiable FL (VFL) is proposed in [74]. This approach used Lagrange interpolation and set interpolation points to verify the integrity of the aggregated gradient. The main advantage of VFL is it enables each participant to verify the aggregated parameters. Moreover, the verification overhead also remains constant regardless of the number of participants. Taking operation and security into account, Zhao *et al.* [68] designed a generic framework of the FL platform by adding a security domain and a cryptographic infrastructure to make trusted connections and interactions among the federated communicating parties. For similar objectives, Ma *et al.* [109] highlighted the most common issues in FL, such as convergence, data poisoning, scaling, model aggregation with security, and privacy perspective and presented potential solutions with simulation results.

A cryptographic approach has been widely adopted as a method of exchanging information and certification to provide security and trust. With the objective of facilitating trusted sharing of cybersecurity certification information following the EU cybersecurity act, work in [55] proposed a generic blockchain platform enriched with smart contract acting as a registry for authoritative device information. The smart contract stores information, such as the manufacturer name, contact information, identity certificate, device type, device id, last firmware version and hash/fingerprint, and a manufacturer usage description (MUD) file describing the typical network interactions and which is published in an off-chain database and others. The proposed blockchain provides a trusted exchange of cybersecurity certification information for any electronic product, service, or process. The authors validated the proposed work by presenting a case study where they used an SDN controller to retrieve a MUD file from the device registry smart contract. To secure communication and data transmission between IoT devices and edge node, Gyamfi *et al.* [45] proposed the elliptic curve cryptography (ECC)-based lightweight cryptographic solution embedded in IoT and edge device. The presented approach consisted of three layers consisting of sensors and actuators (layer I), IoT edge (layer II), and cloud (layer II) where most of the computation including key generation takes place in layer II to reduce computation overhead to the IoT-edge. The IoT-edge layer extracts the public key sent by the server and updates to IoT devices when required. The proposed approach was simulated by configuring IoT edge and docker and the observed results demonstrated reduced running time of encryption as well as reduced resource demands. VerifyNet [62] utilizes a

[1] Homomorphic encryption is a special form of encryption that allows specific types of operations to be done directly on encrypted data without requiring a decryption key. The encrypted result when decrypted, confirms the result of operations performed on the plaintexts [107].

[2] Blockchain technology is a decentralized distributed network that uses public-key cryptography, distributed digital ledger, and consensus algorithms as core components for creating a secure, transparent, and auditable network to allow people/devices to communicate in a trustless manner without presence of any intermediaries [108].

key sharing strategy and encryption to protect the privacy of the user's local gradients in the workflow. Furthermore, this model used the CNN network with the Modified National Institute of Standards and Technology (MNIST) database to test the classification accuracy of the model. The model classifies the correctness of the results returned by the server. Additionally, it also allows users to be offline during the training process.

The cloud service-based architecture is the necessary as well as dominant computing services in today's world. The operations and communications associated with the service provider must be secure and trustworthy. To assess the security and reputation of the cloud service-based architecture for IoT, Li *et al.* [66] proposed a novel trust assessment framework. The proposed framework integrated security and reputation-based trust assessment methods to evaluate the trust of cloud services. Customers' feedback rating for the cloud service's trustworthiness or quality of service of cloud service was incorporated in the framework. For the performance evaluation, the assessment framework was built and tested in two parts, namely, security-based test assessment (SeTA) and reputation-based test assessment (ReTA). SeTA was tested using a synthesized data set encapsulating security metrics whereas ReTA was tested against the WSDream dataset2; a real-world Web service data set and the results demonstrated that the proposed framework efficiently and effectively assesses the trustworthiness of a cloud service while outperforming other trust assessment methods.

A secure data collaboration framework (FDC) consisting of a private data center, public data center, and blockchain technology for the IoT environment was presented in [76]. The role of the private data center is to handle data governance, data registration, and data management where that of the public data center is to facilitate multiparty secure computation. Blockchain technology was used to provide auditable multiparty interactions. The framework was implemented in FL setting to address issues, such as secure and confidential storage, secure sharing and efficient management, traceability and audit of data behaviors, efficient authorization, and others. In another example, PriModChain [77] combined differential privacy-enabled FL, blockchain, and smart contract to ensure privacy, security, reliability, safety, and resiliency in the IIoT environment.

To fully protect the privacy of end users, secure multiparty computation (MPC)[3] approach has also been utilized in FL. Fereidooni *et al.* [111] used MPC to perform secure FL aggregation where the aggregating server(s) cannot access clients' model updates as well as any intermediate global model. To exchange the model update securely, clients use a multiparty encryption scheme to encrypt their updates. Furthermore, to access the global model, the clients decrypt global updates using its secret share of key. After training, clients encrypt their local updates and send it to the server for aggregation.

Despite the several research efforts to make FL secure from attackers controlling end devices and/or acting in the middle, FL can still be vulnerable to centralized server's malfunctioning. Attackers may compromise the aggregating server or server itself may act maliciously. A biased server may manipulate the aggregation process and favor some clients. Considering these possibility, some research (e.g., [112] and [113]) have suggested to use the blockchain technology and delegate all the FL operations to end devices so as to remove the centralized server. By this approach, end devices acting as the miners of the blockchain network collect the model updates, verify it, and finally, perform aggregation. This approach addresses several security concerns but still fails to address the scenario when the client itself can be malicious. Furthermore, the blockchain approach associates high computation and communication requirements and so, it may not be applicable if the end devices are resource constraint.

Securing FL fully is a huge challenge and it is still an open research topic. Cryptographic approaches are quite useful to exchange model updates securely and preserve privacy, however, if the privacy of clients is fully preserved (even to the aggregating server), it is hard to detect malicious model updates and take appropriate measures against colluding attacks. One approach is not sufficient to address all the security concerns associated with FL. Exploring the combination of different approaches discussed above is likely to be a potential solution to address the security issues present in FL.

IV. Resource Constraint, Communication Latency, and Model Accuracy

We have already witnessed the success of blockchain in recent times due to its decentralized model of secure computing. In a similar sense, FL research is growing enormously due to its privacy-preserving decentralized learning model. However, the true success of FL depends on its core challenges, and these need to be addressed for its applicability. The FL framework not only needs to be secure but also should be efficient and accurate enough. The core challenges that hinder the performance of FL are expensive communication, systems heterogeneity, and statistical heterogeneity. In this section, we discuss several research that have addressed such challenges.

In the FL setting, updated model parameters are exchanged regularly between end devices and a central server and it causes a major bottleneck in the performance of federated networks. To alleviate such communication overhead and reduce latency, approaches, such as compression, e.g., [39], clustering, e.g., [40], optimizing global federating learning, e.g., [26], time, and others, have been examined in the literature. The approach to reduce latency might affect the accuracy of the learning model. Several works have also addressed preserving or improving accuracy and in most cases, the accuracy of the proposed solutions has been verified by comparing them with the centralized model.

To alleviate communication overhead in FL, Sattler *et al.* [39] envisioned a compression approach and proposed a new sparse ternary compression (STC) framework. This framework is created by extending the existing compression technique of top-k gradient sparsification. The authors employed a mechanism to enable downstream compression as ternarization and optimal Golomb encoding. The

[3] Secure multiparty computation is a cryptographic protocol that enables distrusting parties to interact and compute a joint function where no individual party can see others' data [110].

authors conducted experiments on the proposed framework by applying four different learning tasks observed that STC performed well in common FL learning scenarios of high-frequency and low-bandwidth communication. Improving communication efficiency by compressing, thus reducing the communicated message size, Lu *et al.* [72] designed and improved the gradient compression algorithm and achieved 8.77% of the original communication time with just 0.03% reduction in the accuracy. This privacy-preserving asynchronous FL mechanism for edge employed collaborative learning of discrete nodes in edge networking with ensuring the privacy of local information. This work also investigated asynchronous FL to better work with diverse characteristics of edge nodes. Preserving accuracy while applying high ratio sparsification in FL, Li *et al.* [70] proposed a general gradient sparsification (GGS) framework for adaptive optimizers. The framework consists of gradient correction and batch normalization up-to-date with local gradients (BN-LG) to keep convergence to a large extent and to minimize the impact of delayed gradients on the training, respectively. Some researchers have addressed communication overhead by tuning the aggregation of the global model. Whereas, Hsieh *et al.* [26] used the approach of aggregating global model only when the global model's weight differs by some empirically selected threshold. With a similar objective and approach as defined in [26], a control algorithm to find global aggregation frequency was proposed in [27]. The control algorithm devised from theoretical analysis learns the system and data characteristics dynamically in real time to find the appropriate aggregation frequency that results in enhancing learning accuracy based on the resource available.

Non-IID data distribution in the FL network is likely to affect the quality of the global model. To address such issue, Yao *et al.* [28] used a feature fusion approach of aggregating local and global model. The proposed model outperformed baselines FL models and demonstrated better accuracy, initialization for new incoming clients, speeding up the convergence process. Wang *et al.* [44] proposed a control algorithm to work with best tradeoff between local update and global parameter aggregation in FL to minimize the loss function under a given resource budget. Considering the effect of statistical heterogeneity, work [29] proposed a novel FMTL framework that forms clusters of clients based on the geometric properties of the FL surface with jointly trainable data distribution. This clustering approach provided better results in the FL scenario where clients' local data is distributed and non-IID. The advantages of this approach compared to the existing methods are that it works with the existing FL communication protocol and is also applicable to general nonconvex objectives. Furthermore, information about a number of clusters does not require to be known in advance.

The clustering approach has also been sought as a solution to address some FL issues. A work presented in [40] proposes a clustering approach to form a cluster among the densely populated devices. A cluster head is then selected and is responsible for enabling self-organizing FL. Battery life, computation resources, and better connectivity (with other devices) parameters were considered for the selection of cluster head. The cluster head then acts as a central server and carries out aggregation task for FL. The authors also presented a heuristic algorithm to optimize global FL time. For quick convergence of the model, work [46] uses a blockchain-based approach to choose a subset of nodes for updating two types of weights in the global model. One subset updates weight based on its local learning accuracy and the other on its participation frequency.

In [36], a federated CLONE model is proposed to work on the edges for connected vehicles network. A parameter EdgeServer was used to coordinate distributed participating vehicles. Each vehicle locally trains its learning model with its own private training data. After one epoch, each vehicle pushes the current value of parameters to the parameter EdgeServer and the EdgeServer aggregates all such parameters from distributed vehicles by computing the weighted average value. For the next epoch, each vehicle pulls the updated parameters as the current parameter from the EdgeServer and repeats the process. In case a new vehicle joins the network, it pulls the current aggregated parameters from the parameter EdgeServer to use as its initial parameters for training. Following asynchronous communication without stopping and waiting for other vehicles to complete an epoch reduces the latency.

System heterogeneity is one of the big issues in the federated network, which cannot be ignored. Ren *et al.* [20] combined the idea of FL and data offloading to alleviate the constraints and challenges of IoT devices. For intensive computation tasks, IoT devices offload data to the edge nodes so that such devices can conserve energy and provide the required quality of service. Multiple deep reinforcement learning (DRL) agents were deployed on IoT devices to assist in offloading decisions as per the dynamic workload and radio environment of the IoT system. DRL agents were trained in a distributed setting using FL and an experiment was conducted to confirm the effectiveness of the edge computing-supported IoT system using data offloading and FL.

Some works incorporated blockchain-based federated model architecture consisting of edge nodes. "FLchain" [65] stores local parameters used for each global aggregation in a block on the channel-specific ledger to enhance security and audit trails. In FLchain, for each new global learning model, a new channel is created. However, the limitations in this model are the blockchain model does not use a reward mechanism for participating nodes, and end devices do not directly participate in BC, in fact, edge devices do all the transactions on behalf of these devices. Moreover, latency of communication, and the computing and storage capability of end devices are not taken into account in the proposed model. Doku and Rawat [73] proposed iFLBC:FL and Blockchain-based ML to bring edge-AI to end devices. To alleviate the scarcity of data, a trained federated shared model is stored in the blockchain that works using the mechanism called Proof of Common Interest (PoCI) to separate relevant and nonrelevant data.

V. Machine Learning Models, Algorithms, and Technology

In this section, we highlight all the ML models, algorithms, and technologies used by surveyed research in Table III. Along with this information, we also present information about the tools and environment under which simulation has been carried

TABLE III
ML MODELS, ALGORITHMS, AND TECHNIQUES USED IN STATE-OF-THE-ART RESEARCH WORKS

Model	FL	ML Models, Algorithms, Technology	Tools and Environment
[43]	✓	SVM	CORE/EMANE Network emulator, TensorFlow
FoolsGold [30]	✓	Softmax classifier, SqueezeNet1.1,	FL prototype using python, VGGNet11
[44]	✓	Squared-SVM, linear regression, K-means, DCNN	Raspberry Pi, Laptops
[49]	✓	SVM, LSTM	virustotal api
PAFLM [72]	✓	three-layer MLP, threshold gradient compression	GPU server,PCs
[31]	✓	IDS	Simulated using Raspberry Pi devices
DeepFed [32]	✓	CNN-GRU,IDS, Paillier cryptosystem	CPU,GPU,Keras API, Flask
FNTAE [50]	✓	KNN	Simulated on workstations
DÏoT [59]	✓	DNN,GRU,IDS	Simulated using IoTs and Gateways
VerifyNet [62]	✓	CNN, Elliptic-Curve	PCs
VFL [74]	✓	Lagrange interpolation, MLP, CNN	Simulated using PCs and Alibaba cloud
[75]	✓	MLPs	Tensorflow, Keras
FDC [76]	✓	DNN, blockchain	Libra, Tensorflow
PriModChain [77]	✓	DNN, Blockchain,Smart contract, Differential privacy	Python, Ethereum, Ganache, Kovan, Scyther
FED-IIoT [79]	✓	GAN	Tensorflow, Keras
[3]	✓	CNN	Simulated at LAN-security Monitoring Project
Gaia [26]	✓	GoogLeNet-CNN,	Amazon-EC2, Emulation-EC2
[27]	✓	SVM, CNN, linear regression, K-means	Simulated using Raspberry Pi and laptops
[52] [109]	✓	CNN	✗
[69]	✓	Q-learning	Ns-3 for mobility
[71]	✓	SAE	LEAF [114]
[28]	✓	CNN	✓
ASTW_FedAVG [41]	✓	CNN,LSTM	Simulated with designed framework
FLchain [65]	✓	Linear regression	✗
[78]	✓	DNN	Simulated with designed framework
STC [39]	✓	sparse ternary compression, LSTM, LR, VGG11	Simulated with designed framework
CLONE [36]	✓	LSTM	Intel FogNode and Jetson TX2
[20]	✓	DRL	IoTs
iFLBC [73]	✓	ML, Blockchain	Simulated with designed framework
[46]	✓	MLP	Simulated with designed framework
DRAFT [48]	✗	-	Fed4FIRE+federated testbeds
[68]	✓	✗	Theoretical concept only
[40]	✓	clustering algorithm	✓
CFL [29]	✓	DCNN, DRNN, clustering	Simulated with designed framework
[70]	✓	CNNs-LeNet-5, DenseNet-121, CifarNet, AlexNet	Simulated with designed framework

out. Our survey is primarily focused on cybersecurity for the IoT environment and importantly using FL. Based on the nature and complexity of the proposed works, authors have adopted a variety of ML models. The only purpose of this section is to give readers information about the trends on kinds of ML models, algorithms, and technologies that have been used by the surveyed works along with the tools and environment under which the proposed works have been evaluated.

For all the proposed works, the authors have adopted varieties of ML models, such as a neural network, SVM, linear regression, *Q*-learning, and so on. FL inherently supports privacy and security (compared to centralized learning), but to strengthen these, some works have also used elliptic-curve cryptography, differential privacy, blockchain, and others. The majority of the works have considered CNNs as their ML models. Different variations of CNNs, such as LeNet, AlexNet, GoogLeNet, VGGNet, and others, have been used. LSTM (a recurrent neural network) and MLPs (a feedforward neural network) also have been used by several works. Several works have adopted multiple of the ML models and compared the results to verify their proposed models.

VI. POPULAR DATA SETS ADOPTED TO EVALUATE LEARNING MODELS

Due to the several challenges associated with IoT and cyberphysical systems as outlined in Section I, these systems have been a primary target of various kinds of cyberattacks in recent times. Because of the huge volume of data flows through the IoT network, data-driven sophisticated anomaly detection systems are necessary for detecting such attacks. A better system needs sufficient high-quality network data to learn the pattern of the compromised network. There have been several works to produce real data set, which can be used to train and test IDS. Moreover, significant efforts also have been devoted to creating data sets to evaluate the performance of FL models. So, in this section, we classify research works based on the data set it uses for their proposed work in Table IV. This classification gives an idea about the most common data sets that have been utilized by several works considered in this article. We also discuss what these data sets are and what they contain so that it might be useful for researchers to choose the data set based on their needs.

TABLE IV
LIST OF DATA SET USED BY VARIOUS RESEARCH WORKS IN THE FIELD OF CYBERSECURITY

Dataset	Dataset used in References	Federated Learning
NSL-KDD [115]	[31], [75]	✓
AWID [116]	[71]	✓
MNIST [117]	[27], [43], [46], [62], [74], [109]	✓
MNIST, Cifar-10 [118]	[72] [29] [28]	✓
MNIST,HAR [119]	[41]	✓
ImageNet	[26]	✓
CIFAR,KWS [120],MNIST	[39]	✓
MNIST, VGGFace2 [121], KDDCup , Amazon reviews [122]	[30]	✓
MNIST, MNIST-F, CIFAR-10	[44]	✓
MNIST, CIFAR-10,ImageNet [123]	[70]	✓
KDD99 [122]	[50]	✓
KDD99 , NSL-KDD	[56]	✗
Mirai [124]	[59]	✓
KDDCup 99 ,NSL-KDD, UNSW-NB15 [80], Kyoto, WSN-DS, CICIDS 2017	[57]	✗
Drebin, Genome, Contagio	[79]	✓
Wearable sensor data collected at kindergarten	[76]	✓
Fed4FIRE+federated testbeds [125]	[48]	✗
KDD, NSLKDD, UNSW-NB15, N-BaIoT	[78]	✓
virustotal api	[49]	✓

KDDCup99 [126] and NSL-KDD [115] are popular intrusion detection data sets, both containing five major intrusion categories listed as follows.

1) *Normal:* No intrusion in the network.
2) *Denial of Service:* Making network resource unavailable by overwhelming it with information and requests.
3) *Remote to User (R2L):* An attack involving unauthorized access to a user machine from a remote machine.
4) *User to Root Attacks (U2R):* Intruder gain access to a network as a legitimate user.
5) *Probe:* Scanning the network to identify weaknesses.

KDDCup99is an intrusion data set created in 1999 with the objective of improving the capability of IDSs. The training set of KDDCup99 contains 3 925 650 attack records and in which only 262 178 records are distinct whereas the test set includes 250 436 attack records and in which only 29 378 records are distinct. In the case of normal traffic data, the training set contains a total of 972 781 records with 812 814 distinct records, and similarly, in the test set, 47 911 records are distinct among 60 591 total records. NSL-KDD is the subset of KDDCup99 created in 2009 to rectify the inefficiencies associated with KDDCup99. The main issue with the KDDCup99 is that it contains significant redundant records, which tend the learning model to be biased toward the more frequent records [115].

Kyoto 2006+ [127] is another NIDS evaluation data set that was produced by processing the data collected from 348 honeypots deployed in five different networks (inside and outside) of Kyoto University. Real as well as virtual machines including two black hole sensors with 318 unused IP addresses were implemented as honeypots to capture the real network traffic data over the three years of span (2006–2009). During this time span, 50 033 015 normal sessions, 42 617 536 known attack sessions, and 425 719 unknown attack sessions were gathered and which were processed further to extract 24 features including 14 derived from the KDDCup99 data set.

VirusTotal API [128] is a cyberthreats scanning service allowing users to analyze files or URL address online. It consists of a large set of analyzers, including antivirus application engines and Website scanners, from more than 60 security vendors. With the VirtusTotal service, users can get a thorough analysis report for submitted files or URLs and if needed, previous analysis reports can also be obtained. The VirusTotal API provides scanning results as a JSON object and with that, an evaluation data set can be developed as required.

AWID [116] is another intrusion data set that comprises real incidents of both normal and anomalous activities that occurred in the 802.11 Wi-Fi networks. Each record in the data set contains 155 attributes with a class attribute for specifying whether the record represents normal or attack traffic. As per the class distribution, AWID has been divided into two major types as a high-level labeled data set (AWID-CLS) and a finer-grained labeled data set (AWID-ATK). AWID-CLS is created from a large set of packets whereas the other is from the smaller subset. These two sets of the data set are formed by capturing packets at different times, in different environment, and with different types of equipment and contain their own set of training and test set. Each record in AWID is classified as either normal or a particular intrusion type. The intrusion types in AWID-CLS are categorized into four major classes named as flooding, impersonation, injection, and normal whereas AWID-ATK specifies more detailed class labeling. The training set of AWID-ATK comprises ten classes whereas a test set contains additional seven classes. The large data set contains 162 375 247 records for training and 48 524 866 records for testing while the reduced data set contains 1 795 575 and 575 643 records for training and testing, respectively, [116].

UNSW-NB15 [80] is another intrusion data set to evaluate NIDSs. The motive behind creating this data set is to mitigate the deficiencies of past intrusion data set and help to identify new and emerging cyberattacks and including low footprint attacks. The UNSW-NB15 data set was created by Australian Centre for Cyber Security (ACCS) that includes real modern as well as synthesized network traffic. A synthesized data set containing both normal and abnormal traffic was created in lab setup using IXIA PerfectStorm tool [129]. This tool contains all the updated publicly known attack information and was used to simulate nine families of attacks named as normal, fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, and worms. Furthermore, other sets of tools and algorithms were also utilized to generate 49 features to cover characteristics of network packets.

WSN-DS [130] is an intrusion data set created for the wireless sensor network (WSN) to train and evaluate IDSs to effectively identify four classes of DoS attacks, namely, blackhole, grayhole, flooding, and scheduling attacks. To collect data for creating WSN-DS, a WSN environment was simulated using network simulator 2 (NS-2) where the LEACH [131] protocol was applied as a routing protocol. The collected data set was then processed and 23 features were produced. The usefulness of the data set was evaluated by training and testing an artificial neural network (ANN).

In another attempt to develop an intrusion data set having the latest threats information and features, the Canadian Institute of Cybersecurity created CICIDS 2017 [54] by collecting five days' network data containing normal and attack traffic in the network environment of the Canadian Institute of Cybersecurity over eight different files. All the files were processed and merged and finally, a single data set fulfilling all the criteria of true intrusion data set was produced. The resultant data set has 2 830 540 records and each record has 83 features including a class label that represents either normal traffic or one of the 14 attack classes.

Mirai actually is not a data set, rather is a worm-like malware that was launched in 2016 [124]. The malware infected distributed IoT devices and transformed them into a botnet, which finally caused one of the most popular DDoS attacks in history. The source of the Mirai attack is publicly available and it is popular among the research community. The source code is launched in an IoT network environment and network traffic is collected and analyzed to create an intrusion data set and moreover, it is also used to evaluate the performance of the developed IDS model.

Fed4FIRE+ [125], a successor of Fed4FIRE, is a project under the European Union's Programme Horizon 2020 started in 2017 with the aim of providing open, accessible, and reliable facilities for supporting experimentally driven research. It provides the largest federation worldwide of next-generation Internet (NGI) testbeds. It aims to support research and innovation communities and initiatives in Europe, including the 5G PPP projects and initiatives. Fed4FIRE+ enables various innovative experiments through the federation of the infrastructures. Moreover, it offers federated hardware and software testbed resources by which an emulation of the network environment can be easily created and cyberattacks experimentation can be conducted efficiently and effectively.

FL research substantially utilizes several ML and DL models and the availability of accessible benchmark data sets allows better training and testing of these models. There have been ample works to create such standard realistic data sets and those have been significantly used in the literature. The MNIST data set [117] is one of the most popular and frequently used of such data sets. It is a simple and most beginner-friendly labeled data set containing 70 000 images of handwritten digits from 0 to 9. There are different variations of MNIST named as MNIST-F and MNIST-O. MNIST-F, which is fashion MNIST, contains a more sophisticated alternative image data set related to ten categories of fashion items. MNIST-F is widely adopted for CNN because of its simplicity to use.

The Canadian Institute For Advanced Research (CIFAR-10) [132] is another image data set consisting of 50 000 training and 10 000 test images categorized over ten classes. MNIST-F contains grayscale images whereas CIFAR-10 is a data set containing color images and is one of the widely used computer-vision data sets for object recognition.

The rapid rise in the availability of multimedia data and enhancement of computing capabilities has assisted on the advancement of building sophisticated and robust ML models. Simple data sets on those sophisticated ML techniques have been no longer useful to identify the true potential of these algorithms. A need for a complex data set is inherent to achieve better results and such necessity led to create ImageNet [123] data set. It is a large-scale data set with high diversity and accuracy compared to most of the existing benchmarked image data sets and is useful mostly for image classification, object localization, and object detection. The data set is a repository of 80 000 synsets of WordNet with an average of 500–1000 clean and full resolution images. The data set has 12 subtrees 3.2 million cleanly annotated images spread over 5247 categories.

VGGFace2 [121] is a large-scale face data set consisting of 3.31 million images of 9131 subjects ranging from a wide range of ethnicities, professions, poses, ages, and illuminations. Google image search was used to download images for all the subjects keeping approximate gender balance. The data set contains images with human-verified bounding boxes around faces and five fiducial keypoints predicted by cascaded CNN. The data set has been partitioned into a training set consisting of 8631 classes and a test set of 500 classes.

The human activity recognition (HAR) [119] data set is a collection of records gathered from activities of daily living (ADL) of 30 subjects where subjects were equipped with a waist-mounted smartphone with embedded inertial sensors. This data set is also publicly available and has been widely used by researchers for activity recognition tasks.

Keyword spotting (KWS) is an activity of identifying keywords from text images, voice commands, and others, however, in this article, we discuss the audio data set [120] used in the research presented in [39]. The data set contains a collection of 105 829 utterances of 35 words of 2168 speakers. Each utterance is stored in the WAVE format file with a length of a maximum of 1 s. The data set is useful widely used for the training and evaluation of speech recognition models.

Amazon reviews [122] data set is produced from a corpus of text in the form of the product reviews by customers on the Amazon commerce Website for authorship identification. The recordset contains 1500 instances with 10 000 attributes and 50 classes. Each record contains attributes related to authors' linguistic style, such as usage of the digit, punctuation, words and sentences' length, usage frequency of words, and so on.

VII. Open Challenges and Future Research Directions

Data are a crucial asset for an individual and company that should be protected to ensure the confidentiality, integrity, and availability (CIA) triad. Legislations, such as the Consumer Data Protection Act and the Data Care Act in the USA, and GDPR in Europe, have been already rolled out to strengthen data protection. However, due to the rapidly growing flood of data, ML has been inevitable to analyze and learn from the data. However, the traditional learning model (centralized) poses a lot of concerns due to the insecure digital highway, limited bandwidth, and sole control of the service provider. In this regard, FL offers an innovative framework to facilitate learning by keeping data locally and training locally. However, it is still in the early stage to be fully applicable particularly for the IoTs environment. In recent times, FL has gained significant attention in the research community. Many works have already proposed their models making use of different ML algorithms, frameworks, and technologies. However, in our survey, we found most of the proposed models use neural networks. NN is mostly preferred in the FL setting, however, it increases the complexity that might increase the overhead in real heterogeneous IoT environments. Moreover, most of the proposed models are simulated in an environment consisting of few devices and that are tested against only a few data sets. To develop an efficient and robust FL model, research works need to consider different permutations and combinations of ML algorithms, data sets, and working dynamics and measure the true efficacy of the developed system.

Considering the limited resources and communication bandwidth in the IoT network, a significant number of research works have proposed an FL scenario where the edge server aggregates the updates from end devices and passes them on to the central server. Such an approach might not work in general as all IoT networks may not have such an ideal configuration. Additionally, the baseline algorithm, FedAvg, has been mostly applied to aggregate and weigh the updated model. Due to the system and statistical heterogeneous characteristics of the IoTs environment, the convergence in real federated networks may not occur as expected. So, it will be valuable to seek other methods that address such issues and result in quick convergence.

Differential privacy, e.g., [23], homomorphic encryption, e.g., [25], and secure function evaluation or multiparty computation, e.g., [24], have been utilized in FL for privacy-preserving learning. FL using these approaches have been implemented and experimented in a small-scale distributed network only. So, it may bring novel challenges in the large-scale network scenarios due to the additional communication and computation burdens.

In the literature, gradient compression schemes (e.g., [39] and [70]) have been popularly applied to compress the communicated messages to thus reducing latency. Although this reduces the size of data to be transmitted, it may result in data loss and affect the accuracy of the learning model.

In surveyed works, the ML learning parameters have been aggregated in a single centralized server. This approach induces the risk of a single point of failure due to a cyberattack or any other reason. Moreover, in this setting, communication efficiency is also likely to be affected by the geographical location of the centralized server. A new approach to design multitier distributed aggregating servers can make FL communication efficient and robust.

Several methods have been proposed to address expensive communication in FL, however, those approaches have been tested only in the small-scale federated networks. Such approaches may perform inefficiently in large-scale federated networks consists of millions of devices with system heterogeneity and statistical heterogeneity. In a large-scale network setting exacerbated by devices sampling and drop out due to network connectivity and limited resources, current approaches are limited to measure the level of system heterogeneity as well as statistical heterogeneity. This deficiency might directly hinder the accuracy of the learning model. Large-scale FL have been highlighted in many articles. These issues have been addressed mostly under the assumptions of i.i.d., nonmodified, and equal data distribution. Identifying and mitigating attacks on true FL stetting without degrading performance and accuracy is still an open area of research.

VIII. Conclusion

In this survey, we first highlighted the risks and threats associated with IoT systems. Motivated by the role of ML to learn from the flood of data and keep the IoT network safe and secure, we talked about different models of learning and pinpointed the merits and demerits of each model. We then extended our study to the application of FL, a new and innovative learning model; for the security of IoT networks. Several recent works addressing the security aspect of IoT environments were discussed. We also discussed several research efforts carried out to mitigate attacks in the FL paradigm. Despite the inherent data protection framework of FL, it bears several challenges to be addressed for its successful adoption. So, we discussed several existing research addressing such performance issues. To assist readers for a research direction with overall information, we presented most of the surveyed works along with the issues addressed and all the ML algorithms, frameworks, technologies, and data sets used by the proposed works. Finally, some open challenges in FL research were presented for future research directions.

Acknowledgment

Any opinion, finding, and conclusions or recommendations expressed in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the funding agencies.

References

[1] S. A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2021.

[2] P. Dube, T. Suk, and C. Wang, "AI Gauge: Runtime estimation for deep learning in the cloud," in *Proc. 31st Int. Symp. Comput. Archit. High Perform. Comput. (SBAC-PAD)*, 2019, pp. 160–167.

[3] Y. Sun, H. Ochiai, and H. Esaki, "Intrusion detection with segmented federated learning for large-scale multiple LANs," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2020, pp. 1–8.

[4] B. Qolomany, K. Ahmad, A. Al-Fuqaha, and J. Qadir, "Particle swarm optimized federated learning for industrial IoT and smart city services," 2020, *arXiv:2009.02560*.

[5] J. Xing, Z. X. Jiang, and H. Yin, "Jupiter: A modern federated learning platform for regional medical care," in *Proc. IEEE Int. Conf. Joint Cloud Comput.*, 2020, p. 21.

[6] A. Jalalirad, M. Scavuzzo, C. Capota, and M. Sprague, "A simple and efficient federated recommender system," in *Proc. 6th IEEE/ACM Int. Conf. Big Data Comput. Appl. Technol.*, 2019, pp. 53–58.

[7] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 46–51, Jun. 2020.

[8] L. U. Khan *et al.*, "Federated learning for edge networks: Resource optimization and incentive mechanism," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 88–93, Oct. 2020.

[9] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara, "Energy demand prediction with federated learning for electric vehicle networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[10] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5341–5351, Aug. 2021.

[11] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[12] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for Internet of Things: Recent advances, taxonomy, and open challenges," 2020, *arXiv:2009.13012*.

[13] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021.

[14] L. Lyu, H. Yu, and Q. Yang, "Threats to federated learning: A survey," 2020, *arXiv:2003.02133*.

[15] Q. Li *et al.*, "A survey on federated learning systems: Vision, hype and reality for data privacy and protection," 2019, *arXiv:1907.09693*.

[16] H. George and A. Arnett, "A case study of implementing cybersecurity best practices for electrical infrastructure in a refinery," in *Proc. IEEE Petroleum Chem. Ind. Committee Conf. (PCIC)*, 2019, pp. 103–108.

[17] T. Choudhury, A. Gupta, S. Pradhan, P. Kumar, and Y. S. Rathore, "Privacy and security of cloud-based Internet of Things (IoT)," in *Proc. 3rd Int. Conf. Comput. Intell. Netw. (CINE)*, 2017, pp. 40–45.

[18] L. Ashiku and C. Dagli, "Cybersecurity as a centralized directed system of systems using SoS explorer as a tool," in *Proc. 14th Annu. Conf. Syst. Syst. Eng. (SoSE)*, 2019, pp. 140–145.

[19] A. Sinaeepourfard, S. Sengupta, J. Krogstie, and R. R. Delgado, "Cybersecurity in large-scale smart cities: Novel proposals for anomaly detection from edge to cloud," in *Proc. Int. Conf. Internet Things Embedded Syst. Commun. (IINTEC)*, 2019, pp. 130–135.

[20] J. Ren, H. Wang, T. Hou, S. Zheng, and C. Tang, "Federated learning-based computation offloading optimization in edge computing-supported Internet of Things," *IEEE Access*, vol. 7, pp. 69194–69201, 2019.

[21] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Artif. Intell. Stat.*, 2017, pp. 1273–1282.

[22] *General Data Protection Regulation (GDPR)*, vol. 1, Intersoft Consult., Fremont, CA, USA, Oct. 2018.

[23] O. Choudhury *et al.*, "Differential privacy-enabled federated learning for sensitive health data," 2019, *arXiv:1910.02578*.

[24] K. Bonawitz *et al.*, "Practical secure aggregation for federated learning on user-held data," 2016, *arXiv:1611.04482*.

[25] K. Cheng, T. Fan, Y. Jin, Y. Liu, T. Chen, and Q. Yang, "Secureboost: A lossless federated learning framework," 2019, *arXiv:1901.08755*.

[26] K. Hsieh *et al.*, "Gaia: Geo-distributed machine learning approaching LAN speeds," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement.*, 2017, pp. 629–647.

[27] S. Wang *et al.*, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 63–71.

[28] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Towards faster and better federated learning: A feature fusion approach," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2019, pp. 175–179.

[29] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3710–3722, Aug. 2021.

[30] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," 2018, *arXiv:1808.04866*.

[31] S. A. Rahman, H. Tout, C. Talhi, and A. Mourad, "Internet of Things intrusion detection: Centralized, on-device, or federated learning?" *IEEE Netw.*, vol. 34, no. 6, pp. 310–317, Nov./Dec. 2020.

[32] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5615–5624, Aug. 2021.

[33] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy, and H. Ming, "AD-IoT: Anomaly detection of IoT cyberattacks in smart city using machine learning," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, 2019, pp. 305–310.

[34] O. Malomo, D. B. Rawat, and M. Garuba, "A federated cloud computing framework for adaptive cyber defense and distributed computing," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 1–6.

[35] O. Malomo, D. Rawat, and M. Garuba, "Security through block vault in a blockchain enabled federated cloud framework," *Appl. Netw. Sci.*, vol. 5, no. 1, pp. 1–18, 2020.

[36] S. Lu, Y. Yao, and W. Shi, "Collaborative learning on the edges: A case study on connected vehicles," in *Proc. 2nd USENIX Workshop Hot Topics Edge Comput. (HotEdge)*, 2019, pp. 1–8.

[37] O. Abdulkader, A. M. Bamhdi, V. Thayananthan, F. Elbouraey, and B. Al-Ghamdi, "A lightweight blockchain based cybersecurity for IoT environments," in *Proc. 6th IEEE Int. Conf. Cyber Security Cloud Comput. (CSCloud)/5th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom)*, 2019, pp. 139–144.

[38] H. Mahdikhani, R. Lu, Y. Zheng, J. Shao, and A. Ghorbani, "Achieving O(log3n) Communication-efficient privacy-preserving range query in fog-based IoT," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5220–5232, Jun. 2020.

[39] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-IID data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3400–3413, Sep. 2020.

[40] L. U. Khan, M. Alsenwi, Z. Han, and C. S. Hong, "Self organizing federated learning over wireless networks: A socially aware clustering approach," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2020, pp. 453–458.

[41] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2020.

[42] J. Kuusijärvi, R. Savola, P. Savolainen, and A. Evesti, "Mitigating IoT security threats with a trusted network element," in *Proc. 11th Int. Conf. Internet Technol. Secured Trans. (ICITST)*, 2016, pp. 260–265.

[43] D. Conway-Jones, T. Tuor, S. Wang, and K. K. Leung, "Demonstration of federated learning in a resource-constrained networked environment," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2019, pp. 484–486.

[44] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[45] E. Gyamfi, J. A. Ansere, and L. Xu, "ECC based lightweight cybersecurity solution for IoT networks utilising multi-access mobile edge computing," in *Proc. 4th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, 2019, pp. 149–154.

[46] Y. J. Kim and C. S. Hong, "Blockchain-based node-aware dynamic weighting methods for improving federated learning performance," in *Proc. 20th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, 2019, pp. 1–4.

[47] F. James, "IoT cybersecurity based smart home intrusion prevention system," in *Proc. 3rd Cyber Security Netw. Conf. (CSNet)*, 2019, pp. 107–113.

[48] S. K. Datta, "DRAFT—A cybersecurity framework for IoT platforms," in *Proc. Zooming Innov. Consum. Technol. Conf. (ZINC)*, 2020, pp. 77–81.

[49] K.-Y. Lin and W.-R. Huang, "Using federated learning on malware classification," in *Proc. 22nd Int. Conf. Adv. Commun. Technol. (ICACT)*, 2020, pp. 585–589.

[50] S. Pang, Y. Peng, T. Ban, D. Inoue, and A. Sarrafzadeh, "A federated network online network traffics analysis engine for cybersecurity," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2015, pp. 1–8.

[51] L. Incipini, A. Belli, L. Palma, R. Concetti, and P. Pierleoni, "MIMIC: A cybersecurity threat turns into a fog computing agent for IoT systems," in *Proc. 42nd Int. Convent. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, 2019, pp. 469–474.

[52] A. M. Zarca, J. B. Bernabe, A. Skarmeta, and J. M. A. Calero, "Virtual IoT honeynets to mitigate cyberattacks in SDN/NFV-enabled IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1262–1277, Jun. 2020.

[53] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020.

[54] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.

[55] R. Neisse, J. L. Hernández-Ramos, S. N. Matheu, G. Baldini, and A. Skarmeta, "Toward a blockchain-based platform to manage cybersecurity certification of IoT devices," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, 2019, pp. 1–6.

[56] J. Shu, L. Zhou, W. Zhang, X. Du, and M. Guizani, "Collaborative intrusion detection for VANETs: A deep learning-based distributed SDN approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4519–4530, Jul. 2020.

[57] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[58] A. Rege *et al.*, "Predicting adversarial cyber-intrusion stages using autoregressive neural networks," *IEEE Intell. Syst.*, vol. 33, no. 2, pp. 29–39, Mar./Apr. 2018.

[59] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A federated self-learning anomaly detection system for IoT," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 756–767.

[60] S. Sen and C. Jayawardena, "Analysis of cyber-attack in big data IoT and cyber-physical systems—A technical approach to cybersecurity modeling," in *Proc. IEEE 5th Int. Conf. Converg. Technol. (I2CT)*, 2019, pp. 1–7.

[61] H. Abie, "Cognitive cybersecurity for CPS-IoT enabled healthcare ecosystems," in *Proc. 13th Int. Symp. Med. Inf. Commun. Technol. (ISMICT)*, 2019, pp. 1–6.

[62] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "VerifyNet: Secure and verifiable federated learning," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 911–926, 2019.

[63] O. Hachinyan, A. Khorina, and S. Zapechnikov, "A game-theoretic technique for securing IoT devices against Mirai botnet," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (EIConRus)*, 2018, pp. 1500–1503.

[64] S. Sen and C. Jayawardena, "Reliability and cybersecurity improvement strategies in wireless sensor networks for IoT-enabled smart infrastructures," in *Proc. Global Conf. Adv. Technol. (GCAT)*, 2019, pp. 1–8.

[65] U. Majeed and C. S. Hong, "FLchain: Federated learning via MEC-enabled blockchain network," in *Proc. 20th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, 2019, pp. 1–4.

[66] X. Li, Q. Wang, X. Lan, X. Chen, N. Zhang, and D. Chen, "Enhancing cloud-based IoT security through trustworthy cloud service: An integration of security and reputation approach," *IEEE Access*, vol. 7, pp. 9368–9383, 2019.

[67] F. Rahman, M. Farmani, M. Tehranipoor, and Y. Jin, "Hardware-assisted cybersecurity for IoT devices," in *Proc. 18th Int. Workshop Microprocessor SoC Test Verification (MTV)*, 2017, pp. 51–56.

[68] L. Zhao, X. Tang, Z. You, Y. Pang, H. Xue, and L. Zhu, "Operation and security considerations of federated learning platform based on compute first network," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC Workshops)*, 2020, pp. 117–121.

[69] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET," *J. Commun. Netw.*, vol. 22, no. 3, pp. 244–258, Jun. 2020.

[70] S. Li, Q. Qi, J. Wang, H. Sun, Y. Li, and F. R. Yu, "GGS: General gradient sparsification for federated learning in edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–7.

[71] B. Cetin, A. Lazar, J. Kim, A. Sim, and K. Wu, "Federated wireless network intrusion detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2019, pp. 6004–6006.

[72] X. Lu, Y. Liao, P. Lio, and P. Hui, "Privacy-preserving asynchronous federated learning mechanism for edge network computing," *IEEE Access*, vol. 8, pp. 48970–48981, 2020.

[73] R. Doku and D. B. Rawat, "IFLBC: On the edge intelligence using federated learning blockchain network," in *Proc. IEEE 6th Int. Conf. Big Data Security Cloud (BigDataSecurity) IEEE Int. Conf. High Perform. Smart Comput. (HPSC) IEEE Int. Conf. Intell. Data Security (IDS)*, 2020, pp. 221–226.

[74] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, and J. Zhang, "VFL: A verifiable federated learning with privacy-preserving for big data in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3316–3326, May 2022.

[75] N. A. A.-A. Al-Marri, B. S. Ciftler, and M. M. Abdallah, "Federated mimic learning for privacy preserving intrusion detection," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, 2020, pp. 1–6.

[76] B. Yin, H. Yin, Y. Wu, and Z. Jiang, "FDC: A secure federated deep learning mechanism for data collaborations in the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6348–6359, Jul. 2020.

[77] P. C. M. Arachchige, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman, "A trustworthy privacy preserving framework for machine learning in industrial IoT systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 9, pp. 6092–6102, Sep. 2020.

[78] T. V. Khoa *et al.*, "Collaborative learning model for cyberattack detection systems in IoT industry 4.0," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2020, pp. 1–6.

[79] R. Taheri, M. Shojafar, M. Alazab, and R. Tafazolli, "Fed-IIoT: A robust federated malware detection architecture in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 12, pp. 8442–8452, Dec. 2021.

[80] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Military Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.

[81] "The SECURED Project (SECURity at the Network EDge)." Webmaster. Jan. 2014. [Online]. Available: https://www.secured-fp7.eu/

[82] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 634–643.

[83] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 739–753.

[84] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Proc. IEEE Symp.Security Privacy (SP)*, 2018, pp. 36–52.

[85] B. Hitaj, G. Ateniese, and F. Perez-Cruz, "Deep models under the GAN: Information leakage from collaborative deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2017, pp. 603–618.

[86] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *Proc. IEEE Symp. Security Privacy (SP)*, 2019, pp. 691–706.

[87] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated Learning*. Cham, Switzerland: Springer, 2020, pp. 17–31.

[88] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," in *Concurrency: The Works of Leslie Lamport*. San Rafael, CA, USA: ACM, 2019, pp. 203–226.

[89] J. R. Douceur, "The Sybil attack," in *Proc. Int. Workshop Peer-to-Peer Syst.*, 2002, pp. 251–260.

[90] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 2938–2948.

[91] H. Wang *et al.*, "Attack of the tails: Yes, you really can backdoor federated learning," 2020, *arXiv:2007.05084*.

[92] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 118–128.

[93] M. Baruch, G. Baruch, and Y. Goldberg, "A little is enough: Circumventing defenses for distributed learning," 2019, *arXiv:1902.06156*.

[94] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5650–5659.

[95] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The hidden vulnerability of distributed learning in Byzantium," 2018, *arXiv:1802.07927*.

[96] L. Li, W. Xu, T. Chen, G. B. Giannakis, and Q. Ling, "RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 1544–1551.

[97] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," 2018, *arXiv:1812.00910*.

[98] C. Xie, O. Koyejo, and I. Gupta, "Zeno: Byzantine-suspicious stochastic gradient descent," 2018, *arXiv:1805.10032*.

[99] G. Sun, Y. Cong, J. Dong, Q. Wang, and J. Liu, "Data poisoning attacks on federated machine learning," 2020, *arXiv:2004.10020*.

[100] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, "Can you really backdoor federated learning?" 2019, *arXiv:1911.07963*.

[101] C. Zhang, S. Li, J. Xia, W. Wang, F. Yan, and Y. Liu, "BatchCrypt: Efficient homomorphic encryption for cross-silo federated learning," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 493–506.

[102] S. Hardy *et al.*, "Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption," 2017, *arXiv:1711.10677*.

[103] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, and R. Zhang, "A hybrid approach to privacy-preserving federated learning," 2018, *arXiv:1812.03224*.

[104] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," 2017, *arXiv:1712.07557*.

[105] L. Zhang, B. Shen, A. Barnawi, S. Xi, N. Kumar, and Y. Wu, "FedDPGAN: Federated differentially private generative adversarial networks framework for the detection of COVID-19 pneumonia," *Inf. Syst. Front.*, vol. 23, pp. 1403–1415, Jun. 2021.

[106] A. Shafee, M. Baza, D. A. Talbert, M. M. Fouda, M. Nabil, and M. Mahmoud, "Mimic learning to generate a shareable network intrusion detection model," in *Proc. IEEE 17th Annu. Consum. Commun. Netw. Conf. (CCNC)*, 2020, pp. 1–6.

[107] X. Yi, R. Paulet, and E. Bertino, "Homomorphic encryption," in *Homomorphic Encryption and Applications*. Cham, Switzerland: Springer, 2014, pp. 27–46.

[108] B. Ghimire and D. B. Rawat, "Secure, privacy preserving and verifiable federating learning using blockchain for Internet of Vehicles," *IEEE Consum. Electron. Mag.*, early access, Jul. 29, 2021, doi: 10.1109/MCE.2021.3097705.

[109] C. Ma *et al.*, "On safeguarding privacy and security in the framework of federated learning," *IEEE Netw.*, vol. 34, no. 4, pp. 242–248, Jul./Aug. 2020.

[110] R. Cramer and I. B. Damgård, *Secure Multiparty Computation*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[111] H. Fereidooni *et al.*, "SAFELearn: Secure aggregation for private federated learning," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2021, pp. 56–62.

[112] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, Aug. 2020.

[113] P. Ramanan and K. Nakayama, "BAFFLE: Blockchain based aggregator free federated learning," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, 2020, pp. 72–81.

[114] S. Caldas *et al.*, "LEAF: A benchmark for federated settings," 2018, *arXiv:1812.01097*.

[115] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP data set," in *Proc. IEEE Symp. Comput. Intell. Security Defense Appl.*, 2009, pp. 1–6.

[116] U. S. K. P. M. Thanthrige, J. Samarabandu, and X. Wang, "Machine learning techniques for intrusion detection on public dataset," in *Proc. IEEE Can. Conf. Electr. Comput. Eng. (CCECE)*, 2016, pp. 1–4.

[117] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[118] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," 2018, *arXiv:1806.00582*.

[119] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. ESANN*, vol. 3, 2013, pp. 437–442.

[120] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," 2018, *arXiv:1804.03209*.

[121] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *Proc. 13th IEEE Int. Conf. Autom. Face Gesture Recognit. (FG)*, 2018, pp. 67–74.

[122] K. Bache and M. Lichman. "UCI Machine Learning Repository." 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[123] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[124] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proc. 26th USENIX Security Symp.*, 2017, pp. 1093–1110.

[125] M. Facca. "Fed4Fire Home." Dec. 2020. [Online]. Available: https://www.fed4fire.eu/

[126] "KDD Cup 1999 Data." 2007. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[127] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation," in *Proc. 1st Workshop Building Anal. Datasets Gathering Exp. Returns Security*, 2011, pp. 29–36.

[128] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Adv. Neural Inf. Process.Syst.* vol. 32, 2019.

[129] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," Dept. Comput. Eng., Chalmers Univ. Technol., Gothenburg, Sweden, Rep. 2000, 2000.

[130] I. Almomani, B. Al-Kasasbeh, and M. Al-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, Sep. 2016, Art. no. 4731953.

[131] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, p. 10.

[132] A. Krizhevsky, V. Nair, and G. Hinton, 2009, "Cifar-10 and Cifar-100 Datasets," [Online]. Available: https://www.cs.toronto.edu/kriz/cifar.html

Bimal Ghimire (Graduate Student Member, IEEE) received the B.E. degree in computer engineering from the Institute of Engineering, Pulchowk Campus, Tribhuvan University, Kirtipur, Nepal, in 2003, and the M.Tech. degree in information technology from Indian Institute of Technology Kharagpur, Kharagpur, India, in 2012. He is currently pursuing the Ph.D. degree in computer science with the Department of Electrical engineering and Computer Science, Howard University, Washington, DC, USA, under the supervision of Prof. D. B. Rawat.

His research interests include cybersecurity, machine learning/federated learning, data analytics, blockchain, Internet of Vehicles, and Internet of Things.

Danda B. Rawat (Senior Member, IEEE) received the Ph.D. degree from Old Dominion University, Norfolk, VA, USA, in 2010.

He is an Associate Dean for Research & Graduate Education, College of Engineering, a Full Professor with the Department of Electrical Engineering and Computer Science, the Founder and the Director of the Data Science and Cybersecurity Center, the Director of DoD Center of Excellence in Artificial Intelligence and Machine Learning, and the Graduate Program Director of Howard CS Graduate Programs, Howard University, Washington, DC, USA. He has secured over 16 million USD in research funding from the U.S. National Science Foundation (NSF), the U.S. Department of Homeland Security (DHS), the U.S. National Security Agency, the U.S. Department of Energy, the National Nuclear Security Administration, DoD and DoD Research Labs, Industry (Microsoft, Intel, and Facebook/Meta.), and private Foundations. He is engaged in research and teaching in the areas of cybersecurity, machine learning, big data analytics, and wireless networking for emerging networked systems, including cyber–physical systems, Internet of Things, multidomain operations, smart cities, software-defined systems, and vehicular networks.

Dr. Rawat is the recipient of the NSF CAREER Award in 2016, the Department of Homeland Security (DHS) Scientific Leadership Award in 2017, the Provost's Distinguished Service Award 2021, the U.S. Air Force Research Laboratory (AFRL) Summer Faculty Visiting Fellowship 2017, and the Best Paper Awards, such as IEEE CCNC, IEEE ICII, and BWCA. He has been serving as an Editor/Guest Editor for over 70 international journals, including the Associate Editor of IEEE TRANSACTIONS OF SERVICE COMPUTING and IEEE TRANSACTIONS OF NETWORK SCIENCE AND ENGINEERING, an Editor of IEEE INTERNET OF THINGS JOURNAL, and the Technical Editors of IEEE NETWORK. He has been in Organizing Committees for several IEEE flagship conferences, such as IEEE INFOCOM, IEEE CNS, and IEEE GLOBECOM. He is a Senior Member of ACM, a member of ASEE and AAAS, and a Fellow of the Institution of Engineering and Technology. He is an ACM Distinguished Speaker in 2021–2023.